

Fermo!Point API

Guida all'integrazione

Versione AsyncAPI: 2.3
Versione EmbedMap: 1.2
Ultimo aggiornamento: 13-06-2018

Autore: Fermopoint Srl

Sommario

Descrizione del documento	3
Il servizio Fermo!Point	4
Integrare Fermo!Point in un e-commerce	4
Integrazione del Workflow ospite	5
Come rendere identificabile il pacco: etichetta di trasporto	5
Introduzione alle API	6
Account Merchant	6
Area Sandbox	6
Metodi di integrazione	6
Soluzioni LIGHT e COMPLETE	6
EmbedMap	6
AsyncAPI	7
Riferimenti utili	7
Guida all'integrazione: AsyncAPI	8
Come effettuare una chiamata	8
Convenzioni	8
Calcolo dell'auth_code	8
Indirizzi di riferimento per l'utilizzo delle API	9
Changelog	9
AsyncAPI Reference	11
Implementazione delle procedure essenziali	11
Workflow base solo Pickup	11
Workflow base Pickup+Delivery	11
Come richiamare le interfacce di AsyncAPI	12
Api pubbliche	12
Api private	12
Interfacce API per utilizzo base	13
Api private: Prenotazione pickup (servizio Pickup)	13
Api private: Spedizioni (servizio Pickup+Delivery)	15
Interfacce Api per utilizzo avanzato (opzionale)	17
Api pubbliche: Negozi (opzionale)	17
Api pubbliche: Utenti (opzionale)	17
Api private: Negozi (opzionale)	18
Api private: Utenti (opzionale)	18
Api private: Merchant (opzionale)	19
Api private: Consegne pickup (opzionale)	20
Api private: Prenotazione pickup (opzionale)	21
Api private: Spedizioni (opzionale)	23
Definizione degli oggetto utilizzati	29
Appendice A - Etichette, DDT e borderò: formati validi per Fermo!Point	36
Etichettatura standard per un pacco con servizio SOLO pickup	36
Etichetta e DDT per il servizio Pickup+Delivery	37
Sistema di automazione tramite Soluzione A+C di BRT	38
In cosa consiste	38
Quando è possibile utilizzare la Soluzione A+C	38
Formato etichette per BRT Soluzione A+C	38
Automatizzare la stampa dell'etichetta	39
Formato valido per etichette ZPL e BRT Sol A+C	39

Descrizione del documento

Il documento descrive le modalità di integrazione del servizio Fermo!Point, i formati validi e le diverse modalità di etichettatura e rappresenta una reference tecnica delle API di integrazione: AsyncAPI.

Legenda

- **F!P:** Fermo!Point
- **Clients/Ecommerce/Merchant:** sono clienti di F!P in possesso di un ecommerce o piattaforma di vendita con backend informatizzato che utilizzano il servizio F!P per il receipt dei pacchi e hanno necessità di automatizzare le spedizioni
- **Utente:** è il cliente finale del merchant
- **Pickup:** si intende il servizio di ritiro per conto terzi svolto dalla rete di negozi F!P
- **Delivery:** si intende il servizio di spedizione tramite corriere fornito da F!P
- **Nickname:** il servizio F!P prevede l'utilizzo di un nickname come "nome" del destinatario presso il negozio scelto. E' necessario (in mancanza del Fermo!Ticket) per identificare il pacco e per segnalare all'utente corretto l'arrivo del pacco
- **Utente ospite:** è un nickname assegnato al merchant per effettuare prenotazioni per utenti ospiti del proprio ecommerce. Viene accompagnato dal merchantId per identificare il nickname ospite di quell'ordine. Es. vendonline (20171010001)
- **Fermo!Ticket:** è il codice identificativo del Pickup per F!P. Un codice alfanumerico di 13 caratteri (es. C6D3CBN534742)
- **pickupId:** altro nome del Fermo!Ticket
- **Id spedizione:** è il codice identificativo di una spedizione con F!P. Il formato è uguale a quello del Fermo!Ticket
- **deliveryId:** altro nome dell'Id spedizione
- **merchantId:** è il codice identificativo di un ordine del merchant, viene fornito dal merchant e deve essere diverso per ogni pickup prenotato. È una stringa lunga al massimo 20 caratteri. Accompagna l'utente ospite del merchant per identificare il nickname ospite di quell'ordine. Es. vendonline (20171010001)

Per gli esempi verranno utilizzati nomi di negozio, ecommerce, utenti e indirizzi puramente casuali e non riferiti ad alcun soggetto reale.

Il servizio Fermo!Point

Fermo!Point (F!P) è il nuovo servizio di **PickUp Point** che sfrutta una rete di negozi selezionati e certificati come recapito dei pacchi per i tuoi clienti.

Un pacco Fermo!Point verrà consegnato a un negozio della nostra rete e riconosciuto come tale grazie all'etichetta applicata sul pacco (vedi [Etichettatura standard per un pacco con servizio SOLO pickup](#)) e confrontando i dati in essa presenti con quelli presenti nella propria consolle di gestione.

Integrare Fermo!Point in un e-commerce

Integrare F!P nel tuo sito e-commerce significa fornire un servizio comodo, sicuro e moderno per i tuoi clienti.

Il servizio funziona tramite l'**acquisto di Fermo!Ticket** (coupon di consegna) che vengono assegnati a un utente del sito e permettono il *ritiro sicuro* della merce presso un nostro negozio abilitato esibendo *nickname* e *codice Fermo!Ticket*.

Per fornire il servizio dal tuo e-commerce dovrai acquistare un certo numero di ticket (secondo le tue esigenze) a prezzo scontato o firmare un contratto di fatturazione mensile. Questi ticket verranno allocati al tuo e-commerce e quando un utente sceglierà F!P per ricevere la tua merce uno di questi gli verrà assegnato. **Il tuo cliente quindi non corrisponderà alcuna cifra a F!P**, pagherà la consegna direttamente a te, al prezzo che riterrai opportuno.

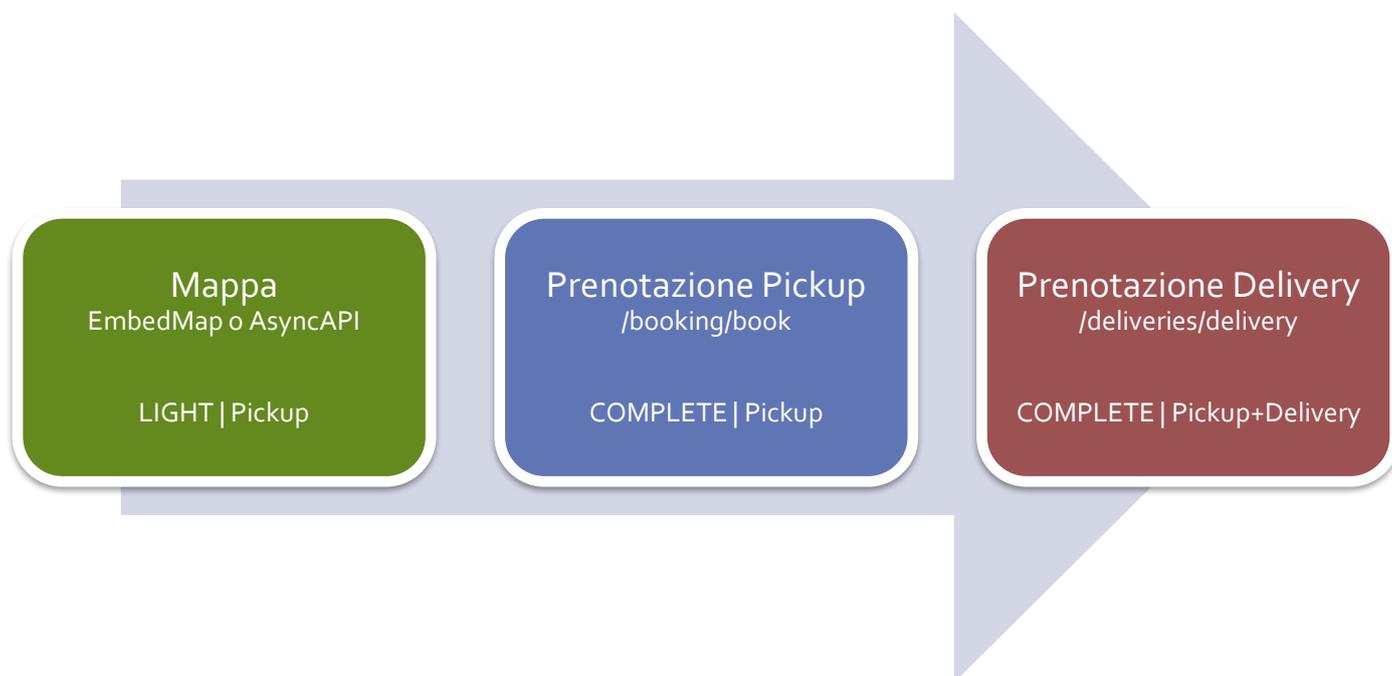
F!P fornisce due tipi di servizio:

1. **Solo Pickup**: prenotazione del solo ritiro presso il negozio scelto, la spedizione è a carico del merchant e avverrà con il suo corriere come se fosse una spedizione normale (salvo l'inserimento dei dati necessari sull'etichetta di trasporto)
2. **Pickup+Delivery**: servizio completo di pickup e spedizione tramite corriere Fermo!Point, che ritirerà la merce nel magazzino del merchant e la consegnerà al point.

Gli step di integrazione fondamentali sono 3:

1. **Mappa**: inserimento della mappa Fermo!Point nel checkout dell'e-commerce
2. **Prenotazione Pickup**: prenotazione tramite AsyncAPI del servizio pickup
3. **Prenotazione Delivery**: prenotazione della spedizione tramite nostro corriere tramite AsyncAPI

In base al servizio scelto e al livello di integrazione (LIGHT o COMPLETE, vedi prossimi paragrafi) saranno da implementare questi semplici step di cui SOLO la MAPPA dovrà essere inserita per forza nel frontend dell'e-commerce.



Integrazione del Workflow ospite

Dal 2018 l'unico flusso supportato dal servizio Fermo!Point è il workflow ospite nel quale il cliente utilizzerà Fermo!Point come **ospite del merchant**.

In questo modo il cliente non dovrà registrarsi a nessun servizio esterno e non verrà quindi interrotto nell'atto dell'acquisto (vedi [Implementazione delle procedure essenziali](#)).

Come rendere identificabile il pacco: etichetta di trasporto

Per rendere identificabile il pacco come consegna Fermo!Point e assegnarlo ad uno specifico cliente è necessario seguire attentamente le linee guida per la compilazione della lettera di vettura.

Attenzione! Se non si rispetta il formato specificato per la lettera di vettura i pacchi potrebbero non giungere al destinatario

Il negoziante che ritira il pacco dal corriere deve in primo luogo capire che si tratta di una consegna Fermo!Point, quindi deve trovare **il logo o la dicitura Fermo!Point (o FP)** da qualche parte sul pacco.

Per riconoscere il cliente a cui consegnare la merce è necessario che l'etichetta applicata sul collo sia **ben visibile** e che **rispetti uno dei due formati** validi (vedi [Etichettatura standard per un pacco con servizio SOLO pickup](#) per ulteriori dettagli):

1. Fermo!Ticket e logo F!P

Il primo formato è quello preferibile se si ha la possibilità di stampare in evidenza il logo Fermo!Point (o la dicitura Fermo!Point) e il Fermo!Ticket.

L'indirizzo di destinazione sarà quello del negozio seguito dal suffisso "- FP".

Attenzione! Questi dati non devono essere coperti da eventuali etichette del corriere, da buste o altro. Si consiglia di applicare l'etichetta su due lati opposti del pacco e controllare che il vettore non copra l'etichetta.

2. Nickname Ospite e dicitura FP

La seconda opzione prevede l'inserimento nell'etichetta del **nickname ospite** come destinatario presso il negozio e un suffisso FP.

OBSOLETO Se non si utilizza il workflow ospite dovrà essere utilizzato al posto del nickname ospite il nickname dell'utente registrato.

Introduzione alle API

Account Merchant

Per utilizzare le API di Fermo!Point (F!P) è necessario possedere un account Merchant.

L'account Merchant viene rilasciato su richiesta dallo staff di F!P. La richiesta può essere fatta dal modulo di contatto sul sito alla pagina <https://www.fermopoint.it/contatti-ecommerce>, inserendo i dati dell'e-commerce da cui si vorrà utilizzare il servizio.

Ad avvenuta registrazione vi verranno inviate le credenziali di riconoscimento: il **client_id** e il **client_secret**.

Il **client_id** è il codice univoco identificativo del vostro account, mentre il **client_secret** è un codice segreto da utilizzare per validare l'account merchant durante le richieste alle API.

In base alla configurazione del vostro contratto vi verranno spediti anche **utente ospite** (per abilitare la prenotazione come ospite) e **account merchant** (per accedere all'area merchant di F!P e alla console di spedizione).

Area Sandbox

Per verificare la correttezza della propria integrazione sarà resa disponibile un'area Sandbox ove sarà possibile accedere con le credenziali di test e provare ad effettuare prenotazioni di consegna su un database separato da quello ufficiale. L'indirizzo ufficiale dell'area sandbox è <https://sandbox.fermopoint.it>.

Le credenziali Sandbox sono condivise con tutti gli altri sviluppatori. Si consiglia di non utilizzare dati sensibili nelle prove.

Di seguito le credenziali Sandbox pubbliche:

```
client_id: peHFRa141xCu4WFnyF8e
client_secret: Q32V3b00nb9c55*JEcb500r54IW23*&LwV70_9IfG56#BK3*
utente Ospite: guest
data di nascita: 2015-01-01
```

Metodi di integrazione

Per integrare il servizio F!P nel vostro e-commerce sono disponibili molteplici soluzioni in base al grado di automazione e alla piattaforma utilizzata.

Questa guida tratta i soli metodi per piattaforme "custom": AsyncAPI e EmbedMap. Per le altre tipologie di integrazione (es. moduli Magento e Prestashop) si consiglia di visitare la pagina:

<https://www.fermopoint.it/business/vantaggi-ecommerce#integrations>

Soluzioni LIGHT e COMPLETE

Sono disponibili più livelli di integrazione del servizio. Per una automazione totale del processo di Pickup o Pickup+Delivery è necessario implementare la soluzione COMPLETE che prevede la prenotazione automatica del pickup nel negozio selezionato e l'eventuale creazione della spedizione per il Pickup+Delivery.

Per completezza riportiamo che anche i moduli Magento e Prestashop implementano COMPLETE.

Se ci fossero problemi di fattibilità, tempi o costi è possibile anche mostrare la sola mappa dei punti (tramite EmbedMap) ed effettuare la prenotazione poi manualmente dai nostri sistemi. Questa modalità è rappresentata dalla soluzione LIGHT.

EmbedMap

Con EmbedMap è possibile mostrare la mappa dei punti Fermo!Point con ricerca tramite servizio Place di Google.

Ogni marker mostra il dettaglio del negozio e un pulsante per la selezione dello stesso a cui è associato un callback javascript nel quale sarà possibile salvare l'ID del punto, utile alla prenotazione successiva tramite AsyncAPI e l'indirizzo del negozio, per salvarlo come destinazione dell'ordine nel vostro e-commerce.

Risulta chiaro quindi che la soluzione EmbedMap implementa in toto la soluzione LIGHT e può essere utilizzata come parte "frontend" della soluzione COMPLETE, associata a un backend che effettua chiamate asincrone/schedulate tramite AsyncAPI.

Incorporare la EmbedMap nel proprio e-commerce è semplicissimo:

1. Incorpora la mappa dei nostri negozi nel tuo e-commerce con un semplice **copia-incolla**
2. Copia l'indirizzo (e/o l'id) del Point scelto **nei tuoi campi di destinazione** standard tramite un callback javascript
3. LIGHT Inserisci manualmente dall'**area merchant** di Fermo!Point le prenotazioni dei tuoi clienti
COMPLETE Utilizza a posteriori l'id del Point salvato per effettuare una prenotazione tramite **AsyncAPI**

Si rimanda al codice scaricabile e alla pagina d'esempio:

Codice scaricabile:

<https://embed.fermopoint.it/Content/modules/light/embed.zip>

Pagina d'esempio:

<https://www.fermopoint.it/Home/TestEmbed>

AsyncAPI

Le AsyncAPI sono sviluppate su un insieme di richieste HTTP GET/POST autenticate con payload formattato in JSON.

Rappresentano il metodo di integrazione di riferimento e permettono di utilizzare qualsiasi servizio F!P all'interno del proprio portale a scapito di un maggior sforzo implementativo per il programmatore. Da queste considerazioni risulta ovvio che le AsyncAPI sono perfette per l'integrazione di una piattaforma su cui si poggiano più portali (ad esempio un gestionale di logistica), ma anche per fornire un'esperienza più fluida al cliente che non dovrà mai "uscire" dal vostro e-commerce.

Con le AsyncAPI oltre alla funzionalità di prenotazione di una consegna, l'e-commerce avrà a disposizione la lista delle prenotazioni effettuate e relativo dettaglio per la parte di backend e la lista dei negozi (point), completa o filtrata per posizione e giorno/ora di apertura, da utilizzare per la visualizzazione su una mappa (es. Google Maps, Bing Maps), i dettagli dei singoli point.

Se non avete necessità particolari, potete utilizzare EmbedMap (widget javascript) per aggiungere la nostra mappa completa di filtro con un semplice copia-incolla.

L'integrazione con le AsyncAPI prevede la creazione di un'interfaccia utente per la scelta del negozio a cui spedire la merce e per l'inserimento dei dati necessari alla prenotazione ed eventualmente alla creazione di un nuovo utente F!P.

Specifichiamo che per motivi di sicurezza tutta la procedura di prenotazione dovrà essere fatta tramite richiesta server-server completa di **client_id** e di un token di controllo generato grazie al **client_secret** (vedere il paragrafo [Come effettuare una chiamata](#)) e che l'e-commerce non verrà mai a conoscenza della password dell'utente F!P, nemmeno in caso di creazione di nuovo utente (la password verrà generata da F!P e spedita all'indirizzo email fornito).

Riferimenti utili

Questa guida è liberamente scaricabile nell'ultima versione aggiornata all'indirizzo:

<https://api.fermopoint.it/Content/files/ApiReference.pdf>

Altri riferimenti utili:

<https://www.fermopoint.it/business/vantaggi-e-commerce#integrations>: Pagina di partenza per quanto riguarda le integrazioni

<http://sandbox.fermopoint.it> : Area sandbox (vedi capitolo [Area Sandbox](#)) e reference online delle AsyncAPI con tool di test

<http://sandbox.fermopoint.it/swagger> : Reference online in formato swagger, utile per test online e per la generazione del codice

<http://sandbox.fermopoint.it/help> : Altra reference online con test online

<http://sandbox.fermopoint.it/Home/Tools>: Strumenti online per lo sviluppatore: calcolo dell'auth_code

<http://ecommercetest.fermopoint.it> : Esempio scaricabile in Microsoft .NET dell'integrazione con Fermo!Point (obsoleto ma utile per la parte di calcolo dell'auth_token e della creazione del pickup)

support@fermopoint.it : Supporto tecnico Fermo!Point, per richiedere supporto o le credenziali di produzione

Guida all'integrazione: AsyncAPI

In questo capitolo verranno elencate le interfacce di integrazione del metodo AsyncAPI.

Per ogni interfaccia verrà proposto del codice di esempio per miglior chiarezza.

Come effettuare una chiamata

Le AsyncAPI sono accessibili tramite richieste HTTP di tipo GET o POST.

Le chiamate pubbliche sono raggiungibili tramite chiamata GET e supportano la specifica Cross-Origin Resource Sharing (CORS) che permette di effettuare le richieste lato client tramite JavaScript.

Per le chiamate private è necessario effettuare richieste utilizzando il metodo POST accompagnato da un unico oggetto JSON come parametro.

Il formato di questo oggetto segue sempre la stessa struttura e comprende i seguenti campi:

- **client_id**: è l'identificativo dell'account comunicato dalla staff di F!P ad attivazione avvenuta.
- **ts**: è il timestamp della richiesta. Non saranno accettati orari che differiscano sensibilmente dall'orario dei nostri server. Fare riferimento al paragrafo Convenzioni per il formato da utilizzare.
- **auth_token**: è il codice di sicurezza da generare a ogni richiesta per validare la richiesta. Vedi Calcolo dell'auth_code.
- **data**: è l'oggetto che contiene i parametri di input specifico della chiamata. [opzionale]

Struttura dell'oggetto JSON che accompagna ogni richiesta POST:

```
{
  "client_id": "stringa identificativa",
  "ts": "timestamp della richiesta in format UTC",
  "auth_token": "stringa generate da client_id, client_secret e ts",
  "data": { "oggetto che contiene i parametri di input di ogni richiesta" }
}
```

Convenzioni

I numeri devono essere specificati senza separatore per le migliaia e utilizzando il punto "." come separatore dei decimali.

Ad esempio il numero 120.783,23 sarà rappresentato da:

```
120783.23
```

Tutte le date e gli orari devono essere specificate secondo la specifica ISO 8601 in Tempo Coordinato Universale (UTC/GMT)

Ad esempio la data 14 ottobre 2014 alle ore 8 minuti 36 secondi 13 e 124 millesimi con fuso orario UTC sarà rappresentata da:

```
2014-10-21T08:03:25.4238996Z o abbreviazioni come 2014-10-21T08:03:25 o 2014-10-21
```

Calcolo dell'auth_code

L'auth_code deve essere calcolato applicando al valore del parametro **ts** della richiesta la funzione HMAC¹ con algoritmo SHA-256 utilizzando come chiave di segreta (**chiave**) la stringa ottenuta concatenando il **client_id** e il **client_secret**.

Di seguito un esempio di calcolo dell'auth_code:

```
ts: 2014-10-21T08:03:25.4238996Z
client_id: peHFRa141xCu4WFnyF8e
client_secret: Q32V3b00nb9c55*JEcbS00r54IW23*&LwV70_9IfG56#BK3*

chiave: peHFRa141xCu4WFnyF8eQ32V3b00nb9c55*JEcbS00r54IW23*&LwV70_9IfG56#BK3*

auth_code = HMAC(ts, chiave, sha-256) = 9722487d44b88c093ea140fc771a43aa2ea6edccea8f40a3bf7e69f6729cd09b
```

¹ HMAC (keyed-hash message authentication code) è un codice per la validazione dell'integrità di un messaggio tramite una funzione di hash e una chiave segreta

Indirizzi di riferimento per l'utilizzo delle API

Ogni versione che verrà rilasciata delle API avrà un indirizzo base differente per garantire, ove possibile, il supporto anche alle versioni precedenti e semplificare la fase di aggiornamento.

API server di produzione:

<https://api.fermopoint.it/api/vX.X>

API server sandbox:

<https://sandbox.fermopoint.it/api/vX.X>

Versione attuale: 2.2

Indirizzo di riferimento per la versione attuale:

Produzione: <https://api.fermopoint.it/api/v2.2>

Sandbox: <http://sandbox.fermopoint.it/api/v2.2>

Changelog

Versione 2.3:

- API per la prenotazione dei resi (/deliveries/return)
- Pulizia della documentazione e spostamento delle API meno usate in un capitolo a parte
- Attivata modifica spedizione per tutti i merchant e creata una specifica per i partner
- Modificata la /deliveries/cancel con la possibilità di cancellare o preservare il pickup collegato

Versione 2.2:

- Aggiunta API per l'aggiornamento delle spedizioni (/deliveries/update)
- Aggiunti custom_label_urls in ApiDelivery
- Aggiunta API per creare una spedizione su un pickup tramite il merchantId (/deliveries/delivery-by-merchant-id)

Versione 2.1:

- Modificate le API di spedizione inserendo i parametri relativi all'indirizzo del magazzino nella richiesta e il nome e contatti del point di destinazione, utile per la creazione di etichette custom (es. ZPL)

Versione 2.0:

- Inserita nuova sezione API spedizioni con prenotazione spedizione (/delivery/delivery e /delivery/cancel)
- Aggiornate nomenclature
- La chiamata /public/points/point/{id} ora ritorna una ApiPointDetails che contiene anche l'url alla pagina del point sul sito FIP
- Aggiunte date di prossima chiusure e riapertura in ApiPoint, ApiPointFull e ApiPointDetails

Versione 1.2:

- Inserita nuova chiamata per ottenere la lista degli ordini semplificata (ticketId, merchantId, stato, ultima modifica) a partire da una data (/orders/states)
- Fix alla documentazione delle interfacce
- Inserito il numero di telefono del point in ApiPointFull (su tutte le versioni)

Versione 1.1:

- Inserita nuova chiamata per verificare la coppia nickname/data di nascita (/public/users/check e /users/check)
- Inserito il conteggio degli ordini nella chiamata /merchant
- Modificata la richiesta della lista degli ordini. Suddivisa in /orders con filtro, paginazione e ordinamento e /orders/last con la lista degli ultimi ordini
- Aggiunta del check sull'utente OSPITE (/users/guest-check)

Versione 1.0:

- Inserita la data di nascita nella prenotazione (/booking/book) per identificare l'utente

- Aggiunta la ricerca dei point nelle API pubbliche
- Aggiunte le chiamate di validazione di nickname e email nelle API pubbliche

Versione 0.9:

- Prima versione pubblica

AsyncAPI Reference

Le AsyncAPI sono divise in interfacce pubbliche e interfacce private.

Le prime possono essere utilizzate da chiunque ad esempio per ottenere la lista dei point da inserire in una mappa sul web. Le Api private, viceversa, possono essere utilizzate solo da chi ha un account merchant attivo e permettono l'integrazione completa con Fermo!Point.

Attenzione: in questa guida verrà indicato per ogni tipologia di chiamata l'indirizzo relativo. Questo deve essere concatenato con l'indirizzo base di produzione o dell'ambiente Sandbox alla versione corretta.

Implementazione delle procedure essenziali

Per semplificare la lettura di questo documento identifichiamo le sole chiamate necessarie all'utilizzo dei workflow Pickup e Pickup+Delivery. Tutte le altre interfacce rappresentano utilità accessorie per implementare altre funzionalità (es. tracciamento, riepilogo ordini, mappa personalizzata, ...)

Il flusso base prevede l'utilizzo della **EmbedMap** per mostrare la mappa nel **frontend** e dare la possibilità di scegliere il negozio. Dal callback della EmbedMap si dovrà salvare l'**id del Point** selezionato, di norma l'id verrà copiato in un campo hidden e poi salvato a database associato all'ordine.

Le altre informazioni necessarie sono una **mail** e un **telefono** (opzionale) del cliente per l'invio delle notifiche del servizio FIP.

Successivamente nel backend o comunque tramite uno scheduler andranno scaricati gli ordini con metodo di spedizione Fermo!Point e andranno fatte le chiamate all'AsyncAPI.

Workflow base solo Pickup

Le uniche chiamate obbligatorie per la prenotazione di un pickup sono la **/booking/book** e la **/booking/approve**.

La prima serve a prenotare il pickup, la seconda a confermarlo. Possono essere effettuate una dopo l'altra se l'ordine è già confermato.

Seguire l'**Esempio 1: Workflow ospite** per la **/booking/book** specificando come **merchantId** (obbligatorio) un codice univoco del vostro ordine (es. il numero d'ordine).

L'**ApiOrder** che viene ritornato dalla chiamata include il Fermo!Ticket del pickup utile al tracciamento della consegna e alla creazione dell'etichetta.

Workflow base Pickup+Delivery

Per prenotare una spedizione con pickup è necessario implementare solamente la chiamata **/deliveries/delivery**

Seguendo l'**Esempio 1: prenotazione di spedizione e consegna pickup in unica operazione** verrà generato automaticamente anche il pickup associato alla spedizione.

Se invece fosse già stato prenotato il pickup tramite la **/booking/book** è possibile prenotare la spedizione partendo dal Fermo!Ticket relativo al pickup seguendo l'**Esempio 2: consegna pickup già prenotata**.

L'**ApiDelivery** che viene ritornata dalla chiamata **/deliveries/delivery** include id della spedizione, Fermo!Ticket del pickup e url dell'etichetta da stampare (o tutti i dati per generare una propria etichetta seguendo le indicazioni in [Etichetta e DDT per il servizio Pickup+Delivery](#)).

Come richiamare le interfacce di AsyncAPI

Api pubbliche

Le chiamate pubbliche non necessitano di payload di autenticazione e possono essere effettuate direttamente lato client tramite JavaScript grazie al supporto di CORS.

```
HTTP GET/POST https://(sandbox|api).fermopoint.it/api/v{versione}/{indirizzo}
```

HEADERS:

Content-Type: application/json

Accept: text/json

Api private

Tutte le chiamate private vanno effettuate tramite metodo POST con payload di autenticazione in JSON:

```
HTTP POST https://(sandbox|api).fermopoint.it/api/v{versione}/{indirizzo}
```

HEADERS:

Content-Type: application/json

Accept: text/json

INPUT:

```
{
  "client_id": "peHFRa141xCu4WFnyF8e",
  "ts": "2014-10-21T08:03:25.4238996Z",
  "auth_token": "9722487d44b88c093ea140fc771a43aa2ea6edccea8f40a3bf7e69f6729cd09b",
  "data": { ... }
}
```

Interfacce API per utilizzo base

Api private: Prenotazione pickup (servizio Pickup)

Prenotazione di una consegna pickup

Descrizione: Prenota una nuova consegna (pickup) utilizzando il **workflow ospite**.

I dati necessari sono le **credenziali** merchant e il **nickname ospite** forniti in fase contrattuale, un codice univoco alfanumerico che sia identificativo dell'ordine generato dall'e-commerce (**merchantId**) e l'**id del negozio** Fermo!Point ottenibile dalla EmbedMap (dovrebbe essere stato salvato nel database collegato all'ordine).

```
Metodo: POST
Indirizzo: /booking/book
Parametri: nessuno
Input: ApiOrderRequest
Output: ApiOrder
```

Esempio: prenotazione pickup con workflow ospite

```
{
  "client_id": "peHFRa141xCu4WFnyF8e",
  "ts": "2014-10-21T08:03:25.4238996Z",
  "auth_token": "9722487d44b88c093ea140fc771a43aa2ea6edccea8f40a3bf7e69f6729cd09b",
  "data": {
    // ApiRequest
    "point_id": "1", // ID del point scelto
    "merchant_id": "3", // ID univoco fornito dal merchant (obbligatorio nel workflow ospite)
    "merchant_notes": "Ordine n.3", // Nota di accompagnamento del merchant
    "existing_user": true,
    "nickname": "vendonline", // Questo è l'utente ospite fornito al merchant
    "born_date": "2016-10-30", // Questa è la data associata all'utente ospite fornito al merchant
    "email": "mario@email.com", // ATTENZIONE, email e numero di telefono per le notifiche dell'ordine
    "phone_number": "333 8778902"
  }
}
```

Risultato: ApiOrder

```
{
  "ticketId": "C032FNA239044",
  "merchant_id": "201507280001",
  "merchant_notes": null,
  "point_id": 1,
  "nickname": "mario",
  "email": "mario@email.com",
  "phone_number": "333 8778902",
  "state": Arrived,
  "notes": [
    {
      "date": "2014-09-05T16:24:37.3127685Z",
      "note": "Arrivato al Fermo!Point"
    }
  ],
  "last_update": "2014-10-17T16:51:05.73Z"
}
```

Conferma di una prenotazione pickup

Descrizione: Conferma una prenotazione precedentemente inserita tramite numero di Fermo!Ticket.

È possibile chiamare la /booking/book e la /booking/approve una dopo l'altra una volta che l'ordine sarà definitivo.

```
Metodo: POST
Indirizzo: /booking/approve/{ticketId}
Parametri: {ticketId}, Numero del Fermo!Ticket associato alla consegna
Input: nessuno
Output: ApiOrder
```

Annullamento di una prenotazione pickup

Descrizione: Annullamento una prenotazione precedentemente inserita tramite numero di Fermo!Ticket. La prenotazione può essere annullata fintantochè il pacco non è arrivato a destinazione (stato Arrived).

```
Metodo: POST
Indirizzo: /booking/cancel/{ticketId}
Parametri: {ticketId}, Numero del Fermo!Ticket associato alla consegna
Input: nessuno
Output: null
```

Altre interfacce per l'utilizzo avanzato sono descritte nella sezione [Interfacce Api per utilizzo avanzato \(opzionale\)](#).

Api private: Spedizioni (servizio Pickup+Delivery)

Prenotazione di una spedizione

Descrizione: Prenota una nuova spedizione con corriere Fermo!Point.

È possibile legare la spedizione a una consegna pickup già prenotata (ad esempio l'e-commerce prenota il pickup e la logistica genera la spedizione collegata a quel pickup).

Alternativamente è possibile prenotare spedizione e pickup insieme specificando l'id del point di destinazione.

Nell'output della chiamata è presente l'url dell'etichetta stampabile in formato HTML.

Attenzione! Il campo `shipping_insurance` deve essere accompagnato da un reale importo della merce (`shipping_insurance_amount`) altrimenti l'assicurazione non sarà valida.

```
Metodo: POST
Indirizzo: /deliveries/delivery
Parametri: nessuno
Input: ApiDeliveryRequest
Output: ApiDelivery
```

Esempio 1: prenotazione di spedizione e consegna pickup in unica operazione

```
{
  "client_id": "peHFRa141xCu4WFnyF8e",
  "auth_token": "9722487d44b88c093ea140fc771a43aa2ea6edccea8f40a3bf7e69f6729cd09b",
  "ts": "2014-10-21T08:03:25.4238996Z",
  "data": {
    "destination_point_id": 19,          // ID del point di destinazione (per la creazione del pickup)

    "merchant_id": "201603140001",      // ID univoco fornito dal merchant
    "merchant_notes": "Ordine n.1",     // Nota di accompagnamento del merchant

    "fullname": "Mario Rossi",
    "email": "mario@email.com",
    "phone_number": "333 8778902",

    "packs": [
      { "weight": 2.0, "size1": 10, "size2": 10, "size3": 10, "notes": "nota pacco 1" },
      { "weight": 2.0, "size1": 10, "size2": 10, "size3": 10, "notes": "nota pacco 2" }
    ],

    "source_warehouse": "MagazzinoEcommerce", // nome magazzino comunicato da F!P (opzionale)

    "shipping_insurance": true,           // assicurare la merce
    "shipping_insurance_amount": 1000.0  // valore merce da assicurare
    "shipping_insurance_is_old": false   // merce nuova o usata
  }
}
```

Esempio 2: consegna con pickup già prenotato

```
{
  "client_id": "peHFRa141xCu4WFnyF8e",
  "auth_token": "9722487d44b88c093ea140fc771a43aa2ea6edccea8f40a3bf7e69f6729cd09b",
  "ts": "2014-10-21T08:03:25.4238996Z",
  "data": {
    "pickup_ticketId": "DD52FZK672117", // ticketId della consegna pickup prenotata

    "fullname": "Mario Rossi",          // nome completo del destinatario
    "email": "mario@email.com",        // email per la notifica
    "phone_number": "333 8778902",     // telefono per la notifica (opzionale)

    "packs": [                          // lista dei pacchi con peso (kg) e dimensioni (cm)
      { "weight": 2.0, "size1": 10, "size2": 10, "size3": 10, "notes": "nota pacco 1" },
      { "weight": 2.0, "size1": 10, "size2": 10, "size3": 10, "notes": "nota pacco 2" }
    ],

    "source_warehouse": "MagazzinoEcommerce", // nome magazzino comunicato da F!P (opzionale)

    "shipping_insurance": true,         // assicurare la merce
    "shipping_insurance_amount": 1000.0 // valore merce da assicurare
    "shipping_insurance_is_old": false  // merce nuova o usata
  }
}
```

Risultato: ApiDelivery (estratto)

```
{
  "deliveryId": "S4B3ACF904026",           // numero della spedizione e codice di tracciamento
  "merchant_id": "201606290001",
  "merchant_notes": "Ordine n.1",

  "source_point_id": null,                 // ID point di partenza (se non c'è un magazzino)
  "source_name": "Magazzino Ecommerce",   // nome magazzino / ragione sociale
  "source_address": {                     // indirizzo magazzino
    ...
  },

  "destination_point_id": 18,
  "destination_name": "Magazzino Ecommerce", // nome point di destinazione
  "destination_address": {                // indirizzo point di destinazione
    ...
  },

  "fullname": "Mario Rossi",
  "email": "info@fermopoint.it",
  "phone_number": null,

  "packs": [
    { "weight": 2.0, "size1": 10, "size2": 10, "size3": 10, "notes": "nota pacco 1" },
    { "weight": 2.0, "size1": 10, "size2": 10, "size3": 10, "notes": "nota pacco 2" }
  ],

  "shipping_insurance": true,              // assicurare la merce
  "shipping_insurance_amount": 1000.0     // valore merce da assicurare
  "shipping_insurance_is_old": false       // merce nuova o usata

  "weight": 4.0,
  "price": 7.7,
  "state": "InDelivery",

  "pickup_order": {                       // pickup collegato (se esiste)
    "ticketId": "DF94EQC112929",
    ...
  },

  "shipping_label_url": "https://...",    // url dell'etichetta di trasporto HTML stampabile
}
```

Annullamento di una spedizione

Descrizione: Annullamento una spedizione precedentemente inserita tramite il “numero di spedizione”. La spedizione può essere annullata fintantochè i pacchi non sono arrivati a destinazione (stato Arrived).

Attenzione! L'eventuale prenotazione pickup collegata verrà Annullata se è impostato il valore `preserve_pickup` a `false`.

```
Metodo: POST
Indirizzo: /deliveries/cancel/{deliveryId}
Parametri: {deliveryId}, identificativo univoco della spedizione (numero di spedizione)
Input: ApiDeliveryCancel
Output: null
```

Esempio: annullamento di una spedizione e del pickup collegato

```
{
  "client_id": "peHFRa141xCu4WFnyF8e",
  "auth_token": "9722487d44b88c093ea140fc771a43aa2ea6edccea8f40a3bf7e69f6729cd09b",
  "ts": "2014-10-21T08:03:25.4238996Z",
  "data": {
    "preserve_pickup": false // se false viene cancellato anche il pickup, se true viene preservato
                             // per essere utilizzato con un'altra spedizione
  }
}
```

Altre interfacce per l'utilizzo avanzato sono descritte nella sezione [Interfacce Api per utilizzo avanzato \(opzionale\)](#).

Interfacce Api per utilizzo avanzato (opzionale)

Api pubbliche: Negozi (opzionale)

Lista dei point

Descrizione: Ritorna la lista di tutti i Point con le informazioni base (nome, posizione) senza i dettagli sugli orari d'apertura

```
Metodo: GET
Indirizzo: /public/points
Parametri: nessuno
Input: nessuno
Output: array di ApiPoint
```

Dettaglio di un singolo point

Descrizione: Dettagli di un Point con tutte le informazioni compresi gli orari d'apertura e l'url alla pagina sul sito di F!P.

```
Metodo: GET
Indirizzo: /public/points/point/{id}
Parametri: {id}, è l'identificativo univoco del point.
Input: nessuno
Output: ApiPointDetails
```

Ricerca dei point

Descrizione: Filtra i Point per posizione (entro un certo raggio) e per giorno/ora di apertura.

ATTENZIONE! Il raggio di ricerca massimo è 50 km. Per raggi superiori è necessario utilizzare le API private.

```
Metodo: POST
Indirizzo: /public/points/search
Parametri: nessuno
Input: ApiSearchData
Output: array di ApiPointFull
```

Api pubbliche: Utenti (opzionale)

Verifica disponibilità nickname

Descrizione: Ritorna TRUE se un nickname è disponibile per la creazione di un nuovo utente

```
Metodo: POST
Indirizzo: /public/users/nickname
Parametri: nessuno
Input: ApiNicknameString
Output: booleano, true se il nickname è disponibile e valido
```

Verifica disponibilità email

Descrizione: Ritorna TRUE se un indirizzo email è disponibile per la creazione di un nuovo utente

```
Metodo: POST
Indirizzo: /public/users/email
Parametri: nessuno
Input: ApiEmailString
Output: booleano, true se l'email è disponibile e valido
```

Verifica dell'utente tramite la coppia nickname e data di nascita

Descrizione: Ritorna TRUE se la coppia nickname e data di nascita corrisponde a un utente registrato

```
Metodo: POST
Indirizzo: /public/users/check
Parametri: nessuno
Input: ApiUserCheck
Output: booleano, true se la coppia nickname e data di nascita è valida
```

Api private: Negozi (opzionale)

Lista dei point con dettaglio

Descrizione: Ritorna la lista di tutti i Point con le informazioni complete dei dettagli sugli orari d'apertura.

ATTENZIONE! Il quantitativo di dati è corposo, si consiglia di salvare la risposta in una cache locale.

```
Metodo: POST
Indirizzo: /points
Parametri: nessuno
Input: nessuno
Output: array di ApiPointFull
```

Ricerca dei point

Descrizione: Filtra i Point per posizione (entro un certo raggio) e per giorno/ora di apertura.

ATTENZIONE! Il quantitativo di dati può essere corposo in caso di raggio troppo elevato (maggiore di 100km), si consiglia eventualmente di salvare la risposta in una cache locale.

```
Metodo: POST
Indirizzo: /points/search
Parametri: nessuno
Input: ApiSearchData
Output: array di ApiPointFull
```

Esempio:

```
{
  "client_id": "peHFRa141xCu4WFnyF8e",
  "ts": "2014-10-21T08:03:25.4238996Z",
  "auth_token": "9722487d44b88c093ea140fc771a43aa2ea6edccea8f40a3bf7e69f6729cd09b",
  "data": {
    // ApiSearchData
    "lat": "45.44",
    "lng": "9.889",
    "radius": 100,
    "day": 0,
    "from": 10,
    "to": 13
  }
}
```

Api private: Utenti (opzionale)

Verifica disponibilità nickname

Descrizione: Ritorna TRUE se un nickname è disponibile per la creazione di un nuovo utente

```
Metodo: POST
Indirizzo: /users/nickname
Parametri: nessuno
Input: ApiNicknameString
Output: booleano, true se il nickname è disponibile e valido
```

Verifica disponibilità email

Descrizione: Ritorna TRUE se un indirizzo email è disponibile per la creazione di un nuovo utente

```
Metodo: POST
Indirizzo: /users/email
Parametri: nessuno
Input: ApiEmailString
Output: booleano, true se l'email è disponibile e valido
```

Verifica dell'utente tramite la coppia nickname e data di nascita

Descrizione: Ritorna TRUE se la coppia nickname e data di nascita corrisponde a un utente registrato

```
Metodo: POST
Indirizzo: /users/check
Parametri: nessuno
Input: ApiUserCheck
Output: booleano, true se la coppia nickname e data di nascita è valida
```

Verifica dell'utente OSPITE dell'e-commerce tramite la coppia nickname e data di nascita

Descrizione: Ritorna TRUE se la coppia nickname e data di nascita corrisponde all'utente attivo come OSPITE dell'e-commerce

```
Metodo: POST
Indirizzo: /users/guest-check
Parametri: nessuno
Input: ApiUserCheck
Output: booleano, true se l'utente è valido ed è attivo come utente OSPITE dell'e-commerce
```

Api private: Merchant (opzionale)

Riepilogo account merchant

Descrizione: Ritorna il numero di Fermo!Ticket in possesso dell'e-commerce e il conteggio delle richieste di prenotazione raggruppate per stato (per i contratti a fatturazione mensile verranno sempre visualizzati 9999 ticket).

```
Metodo: POST
Indirizzo: /merchant
Parametri: nessuno
Input: nessuno
Output: ApiSummary
```

Api private: Consegne pickup (opzionale)

Lista delle consegne prenotate, con filtro, ordinamento e paginazione

Descrizione: Ritorna la lista delle ultime consegne prenotate

```
Metodo: POST
Indirizzo: /orders
Input: ApiOrderFilter
Output: ApiOrderList
```

Lista degli ordini semplificati dell'ecommerce da una data di partenza

Descrizione: Restituisce la lista degli ordini (TicketId, MerchantId, Stato e Ultima modifica) a partire da una data di partenza.

```
Metodo: POST
Indirizzo: /orders/light
Input: ApiLightOrderFilter
Output: ApiLightOrderList
```

Lista delle ultime consegne prenotate

Descrizione: Ritorna la lista delle ultime consegne prenotate

```
Metodo: POST
Indirizzo: /orders/last?take={take}
Parametri:
- {take}, Numero di ordini da ritornare (default 50)
Input: nessuno
Output: array di ApiOrder
```

Dettaglio di una consegna pickup prenotata

Descrizione: Ritorna la consegna associata al numero di Fermo!Ticket fornito.

```
Metodo: POST
Indirizzo: /orders/order/{ticketId}
Parametri: {ticketId}, Numero del Fermo!Ticket associato alla consegna
Input: nessuno
Output: ApiOrder
```

Dettaglio di una consegna pickup prenotata da MerchantId

Descrizione: Ritorna la consegna associata al merchantId fornito.

```
Metodo: POST
Indirizzo: /orders/order-by-merchant-id/{merchantId}
Parametri: {merchantId}, identificativo univoco dell'ordine del merchant associato alla consegna
Input: nessuno
Output: ApiOrder
```

Api private: Prenotazione pickup (opzionale)

Prenotazione di una consegna pickup

Descrizione: Prenota una nuova consegna passando eventualmente i dati per la creazione di un nuovo utente.

Consigliamo di utilizzare solamente il **workflow ospite** (vedi [Interfacce API per utilizzo base](#)). In questo caso non sarà necessario appesantire l'interfaccia utente del checkout, né fa inserire alcun dato al cliente finale: dovrà solo scegliere il punto dalla mappa.

Eventualmente se non richiedi in un altro punto del checkout sarà necessario l'inserimento di una mail e un numero di telefono (opzionale) per le notifiche del servizio.

In definitiva il workflow ospite è risultato il più apprezzato dai nostri clienti.

```
Metodo: POST
Indirizzo: /booking/book
Parametri: nessuno
Input: ApiOrderRequest
Output: ApiOrder
```

OBSOLETO Esempio 2: workflow nuovo utente, passaggio dati per la registrazione a Fermo!Point

```
{
  "client_id": "peHFRa141xCu4WFnyF8e",
  "ts": "2014-10-21T08:03:25.4238996Z",
  "auth_token": "9722487d44b88c093ea140fc771a43aa2ea6edccea8f40a3bf7e69f6729cd09b",
  "data": {
    // ApiRequest
    "point_id": "1", // ID del point scelto
    "merchant_id": "3", // ID univoco fornito dal merchant
    "merchant_notes": "Ordine n.3", // Nota di accompagnamento del merchant
    "existing_user": false,
    "user": { // viene specificata l'entità ApiUser per la creazione di un nuovo utente
      "nickname": "mario",
      "email": "mario@email.com",
      "phone_number": "333 8778902",
      "full_name": "Mario Bianchi",
      "born_date": "1988-01-31",
      "address": {
        "s": "via Antonio Locatelli, 10",
        "e": null,
        "c": "Trezzo d'Adda",
        "l": null,
        "p": "24051",
        "d": "MI"
      },
      "tax_code": "AAAAAA11A11A111A",
      "newsletter": true
    },
    "nickname": "mario", // ATTENZIONE, questo nickname verrà ignorato in caso di nuovo utente
    "born_date": "1988-01-31", // ATTENZIONE, questa borndate verrà ignorata in caso di nuovo utente
    "email": "mario@email.com", // ATTENZIONE, email e numero di telefono per le notifiche dell'ordine
    "phone_number": "333 8778902" // possono differire da quelli forniti alla registrazione dell'utente
  }
}
```

OBSOLETO Esempio 3: workflow utente già registrato a Fermo!Point

```
{
  "client_id": "peHFRa141xCu4WFnyF8e",
  "ts": "2014-10-21T08:03:25.4238996Z",
  "auth_token": "9722487d44b88c093ea140fc771a43aa2ea6edccea8f40a3bf7e69f6729cd09b",
  "data": {
    // ApiRequest
    "point_id": "1", // ID del point scelto
    "merchant_id": "3", // ID univoco fornito dal merchant
    "merchant_notes": "Ordine n.3", // Nota di accompagnamento del merchant
    "existing_user": true,
    "nickname": "mario", // ATTENZIONE, l'utente deve essere registrato
    "born_date": "1988-01-31", // ATTENZIONE, la data deve essere identica a quella nel database
    "email": "mario@email.com", // ATTENZIONE, email e numero di telefono per le notifiche dell'ordine
    "phone_number": "333 8778902"
  }
}
```

Conferma di una prenotazione pickup tramite MerchantId

Descrizione: Conferma una prenotazione precedentemente inserita tramite MerchantId (alternativa all'utilizzo della [Conferma di una prenotazione pickup tramite ticketId](#)).

Attenzione! Il merchantId passato dall'e-commerce deve essere univoco per poter utilizzare questa procedura.

```
Metodo: POST
Indirizzo: /booking/approve-by-merchant-id/{merchantId}
Parametri: {merchantId}, identificativo univoco dell'ordine del merchant associato alla consegna
Input: nessuno
Output: ApiOrder
```

Annullamento di una prenotazione pickup tramite MerchantId

Descrizione: Annullamento una prenotazione precedentemente inserita tramite MerchantId (alternativa all'utilizzo della [Annullamento di una prenotazione pickup tramite ticketId](#)).

La prenotazione può essere annullata fintantochè il pacco non è arrivato a destinazione (stato Arrived).

Attenzione! Il merchantId passato dall'e-commerce deve essere univoco per poter utilizzare questa procedura.

```
Metodo: POST
Indirizzo: /booking/cancel-by-merchant-id/{merchantId}
Parametri: {merchantId}, identificativo univoco dell'ordine del merchant associato alla consegna
Input: nessuno
Output: null
```

Api private: Spedizioni (opzionale)

Lista delle spedizioni prenotate, con filtro, ordinamento e paginazione

Descrizione: Ritorna la lista delle ultime spedizioni prenotate

```
Metodo: POST
Indirizzo: /deliveries
Input: ApiDeliveryFilter
Output: ApiDeliveryList
```

Dettaglio di una spedizione prenotata

Descrizione: Restituisce i dettagli della spedizione corrispondente al codice Fermo!Ticket.

```
Metodo: POST
Indirizzo: /deliveries/delivery/{deliveryId}
Parametri: {deliveryId}, ID della spedizione
Input: nessuno
Output: ApiDelivery
```

Dettaglio di una spedizione da MerchantId

Descrizione: Restituisce i dettagli della spedizione corrispondente al merchantId.

```
Metodo: POST
Indirizzo: /deliveries/delivery-by-merchant-id/{merchantId}
Parametri: {merchantId}, identificativo univoco dell'ordine del merchant associato alla spedizione
Input: nessuno
Output: ApiDelivery
```

Dettaglio di una spedizione da Fermo!Ticket (pickup ticketId)

Descrizione: Restituisce i dettagli della spedizione collegata al ticket (del pickup).

```
Metodo: POST
Indirizzo: /deliveries/delivery-by-ticket-id/{ticketId}
Parametri: {ticketId}, Codice Fermo!Ticket del Pickup associato alla spedizione
Input: nessuno
Output: ApiDelivery
```

Prenotazione di una spedizione

Descrizione: Prenota una nuova spedizione con corriere Fermo!Point.

È possibile legare la spedizione a una consegna pickup già prenotata (ad esempio l'e-commerce prenota il pickup e la logistica genera la spedizione collegata a quel pickup).

Alternativamente è possibile prenotare spedizione e pickup insieme specificando l'id del point di destinazione.

Il terzo esempio mostra una spedizione point to point, al posto di specificare l'indirizzo di ritiro dei pacchi, secondo il contratto stipulato, è possibile selezionare un negozio per il ritiro.

Nell'output della chiamata è presente l'url dell'etichetta stampabile in formato HTML.

Se è attivo il sistema di automazione delle spedizioni di BRT (vedi [Sistema di automazione tramite Soluzione A+C di BRT](#)) saranno presenti i campi:

- brt_label_details: da porre in alto e in evidenza dei DDT o del borderò da consegnare al corriere
- per ogni collo:
 - barcode_text: testo del codice a barre relativo al singolo collo (da riportare in code128)
 - barcode_code128_base64: la codifica in base 64 della stringa già in formato code128 utilizzabile per scrivere il barcode con un font code128

Attenzione! Il campo *shipping_insurance* deve essere accompagnato da un reale importo della merce (*shipping_insurance_amount*) altrimenti l'assicurazione non sarà valida.

```
Metodo: POST
Indirizzo: /deliveries/delivery
Parametri: nessuno
Input: ApiDeliveryRequest
Output: ApiDelivery
```

Esempio 2: consegna pickup già prenotata

```
{
  "client_id": "peHFRa141xCu4WFnyF8e",
  "auth_token": "9722487d44b88c093ea140fc771a43aa2ea6edccea8f40a3bf7e69f6729cd09b",
  "ts": "2014-10-21T08:03:25.4238996Z",
  "data": {
    "pickup_ticketId": "DD52FZK672117", // ticketId della consegna pickup prenotata

    "fullname": "Mario Rossi", // nome completo del destinatario
    "email": "mario@email.com", // email per la notifica
    "phone_number": "333 8778902", // telefono per la notifica (opzionale)

    "packs": [ // lista dei pacchi con peso (kg) e dimensioni (cm)
      { "weight": 2.0, "size1": 10, "size2": 10, "size3": 10, "notes": "nota pacco 1" },
      { "weight": 2.0, "size1": 10, "size2": 10, "size3": 10, "notes": "nota pacco 2" },
      { "weight": 2.0, "size1": 10, "size2": 10, "size3": 10, "notes": "nota pacco 3" }
    ],

    "source_warehouse": "MagazzinoEcommerce", // nome magazzino comunicato da F!P

    "shipping_insurance": true, // assicurare la merce
    "shipping_insurance_amount": 1000.0 // valore merce da assicurare
    "shipping_insurance_is_old": false // merce nuova o usata
  }
}
```

Esempio 3: point to point

```
{
  "client_id": "peHFRa141xCu4WFnyF8e",
  "auth_token": "9722487d44b88c093ea140fc771a43aa2ea6edccea8f40a3bf7e69f6729cd09b",
  "ts": "2014-10-21T08:03:25.4238996Z",
  "data": {
    "source_point_id": 18,           // ID del point di partenza (dove verrà ritirata la merce)
    "destination_point_id": 19,     // ID del point di destinazione (per la creazione del pickup)

    "merchant_id": "201603140001",
    "merchant_notes": "Ordine n.1",

    "fullname": "Mario Rossi",
    "email": "mario@email.com",
    "phone_number": "333 8778902",

    "packs": [
      { "weight": 2.0, "size1": 10, "size2": 10, "size3": 10, "notes": "nota pacco 1" },
      { "weight": 2.0, "size1": 10, "size2": 10, "size3": 10, "notes": "nota pacco 2" },
      { "weight": 2.0, "size1": 10, "size2": 10, "size3": 10, "notes": "nota pacco 3" }
    ],

    "shipping_insurance": true,           // assicurare la merce
    "shipping_insurance_amount": 1000.0  // valore merce da assicurare
    "shipping_insurance_is_old": false    // merce nuova o usata
  }
}
```

Prenotazione di una spedizione presso un negozio con rilocalizzazione automatica al negozio più vicino in caso di chiusura

Descrizione: Prenota una nuova spedizione con corriere Fermo!Point presso il negozio scelto, oppure in caso fosse chiuso per ferie o non più disponibile, la spedizione verrà rilocalata presso il negozio più vicino.

Verrà inviato un ALERT alla mail passata nel campo `relocate_alert_email`. Sarà vostra cura comunicare al cliente il cambio di negozio.

Questa chiamata ritornerà un errore solo in caso di utilizzo di un ID negozio non esistente.

```
Metodo: POST
Indirizzo: /deliveries/delivery-nearby
Parametri: nessuno
Input: ApiDeliveryNearbyRequest
Output: ApiDelivery
```

Esempio:

```
{
  "client_id": "peHFRa141xCu4WFnyF8e",
  "auth_token": "9722487d44b88c093ea140fc771a43aa2ea6edccea8f40a3bf7e69f6729cd09b",
  "ts": "2014-10-21T08:03:25.4238996Z",
  "data": {
    "relocate_alert_email": "alert@merchant.com", // Email alla quale verrà inviato l>alert

    "destination_point_id": 19,           // ID del point di destinazione (per la creazione del pickup)

    "merchant_id": "201603140001",       // ID univoco fornito dal merchant
    "merchant_notes": "Ordine n.1",     // Nota di accompagnamento del merchant

    "fullname": "Mario Rossi",
    "email": "mario@email.com",
    "phone_number": "333 8778902",

    "packs": [
      { "weight": 2.0, "size1": 10, "size2": 10, "size3": 10, "notes": "nota pacco 1" },
      { "weight": 2.0, "size1": 10, "size2": 10, "size3": 10, "notes": "nota pacco 2" }
    ],

    "source_warehouse": "MagazzinoEcommerce", // nome magazzino comunicato da F!P (opzionale)

    "shipping_insurance": true,           // assicurare la merce
    "shipping_insurance_amount": 1000.0  // valore merce da assicurare
    "shipping_insurance_is_old": false    // merce nuova o usata
  }
}
```

Prenotazione di una spedizione da MerchantId

Descrizione: Prenota una nuova spedizione con corriere Fermo!Point collegando un pickup già esistente tramite il passaggio del solo MerchantId.

Attenzione! Il merchantId deve essere UNIVOCO, in caso contrario potrebbero verificarsi degli errori.

Se ci fosse già una spedizione con lo merchantId la chiamata andrà in errore.

Se ci fossero più pickup con lo stesso merchantId verrà utilizzato uno di questi in modo totalmente randomico, quindi la chiamata potrà andare in errore.

```
Metodo: POST
Indirizzo: /deliveries/delivery-by-merchant-id
Parametri: nessuno
Input: ApiDeliveryByMerchantIdRequest
Output: ApiDelivery
```

Esempio: consegna pickup già prenotata tramite MerchantId

```
{
  "client_id": "peHFRa141xCu4WFnyF8e",
  "auth_token": "9722487d44b88c093ea140fc771a43aa2ea6edccea8f40a3bf7e69f6729cd09b",
  "ts": "2014-10-21T08:03:25.4238996Z",
  "data": {
    "merchantId": "201603140001",          // merchantId della consegna pickup già prenotata

    "packs": [                          // lista dei pacchi con peso (kg) e dimensioni (cm)
      { "weight": 2.0, "size1": 10, "size2": 10, "size3": 10, "notes": "nota pacco 1" },
      { "weight": 2.0, "size1": 10, "size2": 10, "size3": 10, "notes": "nota pacco 2" },
      { "weight": 2.0, "size1": 10, "size2": 10, "size3": 10, "notes": "nota pacco 3" }
    ],

    "source_warehouse": "MagazzinoEcommerce", // nome magazzino comunicato da F!P

    "shipping_insurance": true,              // assicurare la merce
    "shipping_insurance_amount": 1000.0     // valore merce da assicurare
    "shipping_insurance_is_old": false      // merce nuova o usata
  }
}
```

Prenotazione di una reso (solo se attivato a contratto)

Descrizione: Prenota un reso con corriere Fermo!Point con partenza da un Point.

```
Metodo: POST
Indirizzo: /deliveries/retudn
Parametri: nessuno
Input: ApiReturnRequest
Output: ApiDelivery
```

Esempio: prenotazione di un reso

```
{
  "client_id": "peHFRa141xCu4WFnyF8e",
  "auth_token": "9722487d44b88c093ea140fc771a43aa2ea6edccea8f40a3bf7e69f6729cd09b",
  "ts": "2014-10-21T08:03:25.4238996Z",
  "data": {
    "source_point_id": 19, // ID del point di partenza
    "destination_warehouse": "MagazzinoEcommerce", // magazzino comunicato da F!P (obbligatorio)

    "return_code": "CodiceReso", // Codice del reso comunicato da F!P (opzionale)

    "merchant_id": "201603140001", // ID univoco fornito dal merchant
    "merchant_notes": "Ordine n.1", // Nota di accompagnamento del merchant

    "fullname": "Mario Rossi",
    "email": "mario@email.com",
    "phone_number": "333 8778902",

    "packs": [
      { "weight": 2.0, "size1": 10, "size2": 10, "size3": 10, "notes": "nota pacco 1" },
      { "weight": 2.0, "size1": 10, "size2": 10, "size3": 10, "notes": "nota pacco 2" }
    ],

    "shipping_insurance": true, // assicurare la merce
    "shipping_insurance_amount": 1000.0 // valore merce da assicurare
    "shipping_insurance_is_old": false // merce nuova o usata
  }
}
```

Aggiornamento di una spedizione

Descrizione: Aggiorna lo stato di una. Questa operazione può essere effettuata fintantochè il pacco non sarà a destinazione.

Attenzione! Verranno addebitate delle penali e subito sospeso il servizio in caso di aggiornamento con dati non veritieri.

```
Metodo: POST
Indirizzo: /deliveries/update/{deliveryId}
Parametri: {deliveryId}, identificativo univoco della spedizione (numero di spedizione)
Input: ApiDeliveryUpdate
Output: ApiDelivery
```

Esempio: aggiornamento dello stato e dei colli di una spedizione

```
{
  "client_id": "peHFRa141xCu4WFnyF8e",
  "auth_token": "9722487d44b88c093ea140fc771a43aa2ea6edccea8f40a3bf7e69f6729cd09b",
  "ts": "2014-10-21T08:03:25.4238996Z",
  "data": {
    "update_packs_size": [
      { "weight": 2.0, "size1": 10, "size2": 10, "size3": 10 },
      { "weight": 2.0, "size1": 10, "size2": 10, "size3": 10 },
      { "weight": 2.0, "size1": 10, "size2": 10, "size3": 10 }
    ]
  }
}
```

Aggiornamento di una spedizione con stato (per partner certificati)

Descrizione: Aggiorna lo stato di una spedizione e pesi e dimensioni dei colli in essa inclusi. Questa operazione è attivata solo per alcuni vettori speciali.

Attenzione! Verranno addebitate delle penali e subito sospeso il servizio in caso di aggiornamento con dati non veritieri.

```
Metodo: POST
Indirizzo: /deliveries/update-by-partner/{deliveryId}
Parametri: {deliveryId}, identificativo univoco della spedizione (numero di spedizione)
Input: ApiDeliveryUpdateByPartner
Output: ApiDelivery
```

Esempio: aggiornamento dello stato e dei colli di una spedizione

```
{
  "client_id": "peHFRa141xCu4WFnyF8e",
  "auth_token": "9722487d44b88c093ea140fc771a43aa2ea6edccea8f40a3bf7e69f6729cd09b",
  "ts": "2014-10-21T08:03:25.4238996Z",
  "data": {
    "updated_state": "InDelivery",
    "updated_packs_size": [
      { "weight": 2.0, "size1": 10, "size2": 10, "size3": 10 },
      { "weight": 2.0, "size1": 10, "size2": 10, "size3": 10 },
      { "weight": 2.0, "size1": 10, "size2": 10, "size3": 10 }
    ]
  }
}
```

Definizione degli oggetto utilizzati

ApiOrderState

```
[
  "Init",                // ordine inizializzato
  "WaitingForPayment",   // in attesa della conferma dall'e-commerce (passo 1)
  "WaitingForPaymentConfirm", // in attesa della conferma dall'e-commerce (passo 2)
  "WaitingForPaymentCapture", // in attesa della conferma dall'e-commerce (passo 3)
  "ToDelivery",         // pronto per la spedizione (solo spedizione)
  "InDelivery",        // in spedizione (etichetta generata, solo spedizione)
  "Booked",            // prenotato
  "Sent",              // non usato dalle api
  "Arrived",          // arrivato al negozio (di partenza per una spedizione)
  "Forwarded",        // consegnato al corriere dal negozio di partenza (solo spedizione)
  "Received",         // arrivato al negozio di destinazione (solo spedizione)
  "Collected",       // ritirato
  "Canceled",         // annullato
  "ToDeposit",        // giacenza scaduta, in ritiro da parte di Fermo!Point
  "InDeposit",        // nel deposito Fermo!Point
  "Error"             // errore
]
```

ApiRequest

```
{
  "point_id": "1",           // ID del point scelto
  "merchant_id": "3",       // ID univoco fornito dal merchant
  "merchant_notes": "Ordine n.3", // Nota di accompagnamento del merchant
  "existing_user": false,    // true se l'utente è già registrato, false se nuovo utente
  "user": { ... },         // entità ApiUser per la creazione di un nuovo utente
  "nickname": "mario",     // ATTENZIONE, questo nickname verrà ignorato in caso di nuovo utente
  "email": "mario@email.com", // ATTENZIONE, email e numero di telefono per le notifiche dell'ordine
  "phone_number": "333 8778902" // possono differire da quelli forniti alla registrazione dell'utente
}
```

ApiOrderSummary

```
{
  "state": "WaitingForPayment", // ApiOrderState
  "count": 1
}
```

ApiSummary

```
{
  "credits": 7,           // Fermo!Ticket disponibili
  "orders_count": 2,     // Numero totale di ordini effettuati
  "orders": [           // ApiOrderSummary
    {
      "state": "WaitingForPayment",
      "count": 1
    },
    {
      "state": "Collected",
      "count": 1
    }
  ]
}
```

ApiLightOrderFilter

```
{
  "date_from": "2014-03-10T17:32:00.3158Z", // filtro data (da)
}
```

ApiOrderFilter

```
{
  "skip": 0, // numero ordini del filtro da saltare
  "take": 10, // numero ordini del filtro da ritornare
  "orderby": "date", // ordinamento*
  "nickname": "", // filtro sul nickname
  "state": "All", // filtro sullo stato (All o ApiOrderState)
  "pointId": null, // filtro sull'ID del negozio
  "merchantId": null, // filtro sul merchantId
  "date_from": "2014-03-10T17:32:00.3158Z", // filtro data (da)
  "date_to": "2016-03-10T17:32:00.3158Z" // filtro data (a)
}
* ordinamento: date (data decrescente), nickname, state, pointid, merchantid
```

ApiDeliveryFilter

```
{
  "skip": 0, // numero ordini del filtro da saltare
  "take": 10, // numero ordini del filtro da ritornare
  "orderby": "date", // ordinamento*
  "state": "All", // filtro sullo stato (All o ApiOrderState)
  "destinationPointId": null, // filtro sull'ID del negozio di destinazione
  "merchantId": null, // filtro sul merchantId
  "date_from": "2014-03-10T17:32:00.3158Z", // filtro data (da)
  "date_to": "2016-03-10T17:32:00.3158Z" // filtro data (a)
}
* ordinamento: date (data decrescente), state, destinationPointid, merchantid
```

ApiOrderList

```
{
  "orders_count": 2, // Numero di ordini non paginati
  "orders": [ ] // Array di ApiOrder paginati
}
```

ApiLightOrderList

```
{
  "orders_count": 2, // Numero di ordini non paginati
  "orders": [ ] // Array di ApiLightOrder paginati
}
```

ApiDeliveryList

```
{
  "deliveries_count": 2, // Numero di spedizioni non paginate
  "deliveries": [ ] // Array di ApiDelivery paginate
}
```

ApiSearchData

```
{
  "lat": "45.44", // latitudine
  "lng": "9.889", // longitudine
  "radius": 100, // raggio in km
  "day": 0, // null = tutti i giorni, 0 = domenica, ... , 6 = sabato
  "from": 10, // intervallo di apertura: da
  "to": 13 // intervallo di apertura: fino a
}
```

ApiNicknameString

```
{
  "nickname": "mario" // nickname richiesto per la verifica della disponibilità
}
```

ApiEmailString

```
{
  "email": "mario@email.com"           // indirizzo email richiesto per la verifica della disponibilità
}
```

ApiUserCheck

```
{
  "nickname": "mario"                  // nickname dell'utente da verificare
  "born_date": "1988-10-01"           // data di nascita dell'utente da verificare
}
```

ApiAddress

```
{
  "s": "via Antonio Locatelli, 10",    // indirizzo
  "e": null,                           // indirizzo seconda linea (opzionale)
  "c": "Trezzo d'Adda",                // città
  "l": null,                           // località (opzionale)
  "p": "24051",                        // Codice di Avviamento Postale (CAP)
  "d": "MI"                            // Provincia (codice identificativo standard di due lettere)
}
```

ApiUser

```
{
  "nickname": "mario",                 // nickname per l'accesso a Fermo!Point da presentare al negoziante
  "email": "mario@email.com",         // email di registrazione e predefinito per le notifiche
  "phone_number": "333 8778902",      // numero telefonico predefinito per notifiche
  "full_name": "Mario Bianchi",       // nome completo
  "born_date": "1988-01-01",          // data di nascita
  "address": { ... },                 // ApiAddress, indirizzo di residenza
  "tax_code": "AAAAAA11A11A111A",    // Codice Fiscale o Partita Iva
  "newsletter": true                  // autorizzazione all'invio di newsletter informative
}
```

ApiPoint

```
{
  "id": 1,                             // identificativo del point
  "n": "Cartoleria nuova stella",       // nome del point
  "lt": 45.5183661,                    // latitudine
  "ln": 9.5920823,                     // longitudine
  "c": "2017-01-01",                   // periodo di chiusura: chiuso da (null se non impostata)
  "r": "2017-01-15"                    // periodo di chiusura: chiuso fino a (null se non impostata)
}
```

ApiOpening

```
{
  "d": 1,                               // giorno della settimana, 0 = domenica - 6 = sabato
  "h": [                                 // array degli orari d'apertura, o = apre, c = chiude, orari in decimale
    { "o": 9.0, "c": 13.0 },
    { "o": 16.0, "c": 19.0 }
  ]
}
```

ApiPointFull

```
{
  "id": 1, // identificativo del point
  "n": "Cartoleria nuova stella", // nome del point
  "lt": 45.5183661, // latitudine
  "ln": 9.5920823, // longitudine
  "c": "2017-01-01", // periodo di chiusura: chiuso da (null se non impostata)
  "r": "2017-01-15", // periodo di chiusura: chiuso fino a (null se non impostata)
  "d": 0, // distanza in km
  "a": { ... }, // ApiAddress indirizzo del point
  "p": "Anna Deda", // proprietario
  "s": "Cartoleria, Merceria", // settore merceologico
  "o": [ // array di ApiOpening, orari d'apertura
    { "d": 1, "h": [ { "o": 9.0, "c": 19.0 } ] },
    { "d": 2, "h": [ { "o": 9.0, "c": 19.0 } ] },
    { "d": 3, "h": [ { "o": 9.0, "c": 19.0 } ] },
    { "d": 4, "h": [ { "o": 9.0, "c": 19.0 } ] },
    { "d": 5, "h": [ { "o": 9.0, "c": 19.0 } ] }
  ],
  "t": "0363 87362" // numero di telefono del point (da non mostrare sul sito)
}
```

ApiPointDetails

```
{
  "id": 1, // identificativo del point
  "n": "Cartoleria nuova stella", // nome del point
  "lt": 45.5183661, // latitudine
  "ln": 9.5920823, // longitudine
  "c": "2017-01-01", // periodo di chiusura: chiuso da (null se non impostata)
  "r": "2017-01-15", // periodo di chiusura: chiuso fino a (null se non impostata)
  "d": 0, // distanza in km
  "a": { ... }, // ApiAddress indirizzo del point
  "p": "Anna Deda", // proprietario
  "s": "Cartoleria, Merceria", // settore merceologico
  "o": [ // array di ApiOpening, orari d'apertura
    { "d": 1, "h": [ { "o": 9.0, "c": 19.0 } ] },
    { "d": 2, "h": [ { "o": 9.0, "c": 19.0 } ] },
    { "d": 3, "h": [ { "o": 9.0, "c": 19.0 } ] },
    { "d": 4, "h": [ { "o": 9.0, "c": 19.0 } ] },
    { "d": 5, "h": [ { "o": 9.0, "c": 19.0 } ] }
  ],
  "t": "0363 87362" // numero di telefono del point (da non mostrare sul sito)
  "u": "https://www.fermopoint.it/negoziario/1" // url alla pagina del negozio sul sito di fermopoint
}
```

ApiOrder

```
{
  "ticketId": "C032FNA239044",
  "merchant_id": "201507280001",
  "merchant_notes": null,
  "point_id": 1,
  "nickname": "mario",
  "email": "mario@email.com ",
  "phone_number": "333 8778902",
  "state": Arrived,
  "notes": [
    {
      "date": "2014-09-05T16:24:37.3127685Z",
      "note": "Arrivato al Fermo!Point"
    }
  ],
  "last_update": "2014-10-17T16:51:05.73Z"
}
```

ApiLightOrder

```
{
  "ticketId": "C032FNA239044",
  "merchant_id": "201507280001",
  "state": Arrived,
  "last_update": "2014-10-17T16:51:05.73Z"
}
```

ApiDeliveryRequest

```
{
  "source_point_id": null,
  "source_address": {
    ...
  },
  "destination_point_id": 19,
  "destination_address": null,
  "merchant_id": "201603140001",
  "merchant_notes": "Ordine n.1",
  "fullname": "Mario Rossi",
  "email": "mario@email.com",
  "phone_number": "333 8778902",
  "packs": [
    { "weight": 2.0, "size1": 10, "size2": 10, "size3": 10, "notes": "nota pacco 1" },
    { "weight": 2.0, "size1": 10, "size2": 10, "size3": 10, "notes": "nota pacco 2" }
  ],
  "custom_label_url": [
    { "type": "pdf", "url": "http://.... " },
    { "type": "zpl", "url": "http://.... " },
    { "type": "jpg", "url": "http://.... " }
  ],
  "shipping_insurance": true,
  "shipping_insurance_amount": 1000.0
  "shipping_insurance_is_old": false
}
```

ApiDeliveryNearbyRequest

```
{
  "source_point_id": null,
  "source_address": {
    ...
  },
  "relocate_alert_email": "alert@merchant.com",
  "destination_point_id": 19,
  "merchant_id": "201603140001",
  "merchant_notes": "Ordine n.1",
  "fullname": "Mario Rossi",
  "email": "mario@email.com",
  "phone_number": "333 8778902",
  "packs": [
    { "weight": 2.0, "size1": 10, "size2": 10, "size3": 10, "notes": "nota pacco 1" },
    { "weight": 2.0, "size1": 10, "size2": 10, "size3": 10, "notes": "nota pacco 2" }
  ],
  "custom_label_url": [
    { "type": "pdf", "url": "http://.... " },
    { "type": "zpl", "url": "http://.... " },
    { "type": "jpg", "url": "http://.... " }
  ],
  "shipping_insurance": true,
  "shipping_insurance_amount": 1000.0
  "shipping_insurance_is_old": false
}
```

ApiReturnRequest

```
{
  "source_point_id": 19,
  "destination_warehouse": "MagazzinoEcommerce",
  "return_code": "CodiceReso",
  "merchant_id": "201603140001",
  "merchant_notes": "Ordine n.1",
  "fullname": "Mario Rossi",
  "email": "mario@email.com",
  "phone_number": "333 8778902",
  "packs": [
    { "weight": 2.0, "size1": 10, "size2": 10, "size3": 10, "notes": "nota pacco 1" }
  ],
  "shipping_insurance": true,
  "shipping_insurance_amount": 1000.0
  "shipping_insurance_is_old": false
}
```

ApiDeliveryByMerchantIdRequest

```
{
  "source_point_id": null,
  "source_address": {
    ...
  },
  "merchant_id": "201603140001",
  "packs": [
    { "weight": 2.0, "size1": 10, "size2": 10, "size3": 10, "notes": "nota pacco 1" },
    { "weight": 2.0, "size1": 10, "size2": 10, "size3": 10, "notes": "nota pacco 2" },
    { "weight": 2.0, "size1": 10, "size2": 10, "size3": 10, "notes": "nota pacco 3" }
  ],
  "custom_label_url": [
    // Utilizzato solo da logistiche avanzate per etichette custom
    { "type": "pdf", "url": "http://.... " },
    { "type": "zpl", "url": "http://.... " },
    { "type": "jpg", "url": "http://.... " }
  ],
  "shipping_insurance": true, // assicurare la merce
  "shipping_insurance_amount": 1000.0 // valore merce da assicurare
  "shipping_insurance_is_old": false // merce nuova o usata
}
```

ApiPackSize

```
{
  "weight": 2.0,
  "size1": 10, "size2": 10, "size3": 10
}
```

ApiPack

```
{
  "weight": 2.0,
  "size1": 10, "size2": 10, "size3": 10,
  "notes": "nota pacco 1",
  "barcode_text": "S4B3ACF904026001", // generato: testo descrittivo codice a barre
  "barcode_code128_base64": "w4xTNEIzQUNGw4d6SDogw4gxw4rDjg==" // generato: base64 della stringa del code128
}
```

ApiLabelUrl

```
{
  "type": "pdf",
  "url": "http://... "
}
```

ApiDeliveryCancel

```
{
  "preserve_pickup": false // se false viene cancellato anche il pickup, se true viene preservato
                          // per essere utilizzato con un'altra spedizione
}
```

ApiDeliveryUpdate

```
{
  "update_packs_size": [
    { "weight": 2.0, "size1": 10, "size2": 10, "size3": 10 },
    { "weight": 2.0, "size1": 10, "size2": 10, "size3": 10 },
    { "weight": 2.0, "size1": 10, "size2": 10, "size3": 10 }
  ]
}
```

ApiDeliveryUpdateByPartner

```
{
  "updated_state": "InDelivery",
  "update_packs_size": [
    { "weight": 2.0, "size1": 10, "size2": 10, "size3": 10 },
    { "weight": 2.0, "size1": 10, "size2": 10, "size3": 10 },
    { "weight": 2.0, "size1": 10, "size2": 10, "size3": 10 }
  ]
}
```

ApiDelivery

```
{
  "deliveryId": "S4B3ACF904026",

  "merchant_id": "201606290001",
  "merchant_notes": "Ordine n.1",

  "source_point_id": null, // ID point di partenza (se non c'è un magazzino)
  "source_name": "Magazzino Ecommerce", // nome magazzino / ragione sociale
  "source_contact": "Nome addetto logistica", // referente in magazzino
  "source_phone_number": "02 1222033", // telefono magazzino
  "source_address": { // indirizzo magazzino
    ...
  },

  "destination_point_id": 18,
  "destination_name": "Magazzino Ecommerce", // nome point di destinazione
  "destination_contact": "Nome addetto logistica", // referente point di destinazione
  "destination_phone_number": "02 1222033", // telefono point di destinazione
  "destination_address": { // indirizzo point di destinazione
    ...
  },

  "fullname": "Mario Rossi",
  "email": "info@fermopoint.it",
  "phone_number": null,

  "packs": [
    { "weight": 2.0, "size1": 10, "size2": 10, "size3": 10, "notes": "nota pacco 1",
      "barcode_text": " S4B3ACF904026001", "barcode_code128_base64": "w4xTNEIzQUNGw4d6SDogw4gxw4rDjg==" },
    { "weight": 2.0, "size1": 10, "size2": 10, "size3": 10, "notes": "nota pacco 2",
      "barcode_text": " S4B3ACF904026002", "barcode_code128_base64": "w4xTNEIzQUNGw4d6SDogw4gyLc00" },
    { "weight": 2.0, "size1": 10, "size2": 10, "size3": 10, "notes": "nota pacco 3",
      "barcode_text": " S4B3ACF904026003", "barcode_code128_base64": "w4xTNEIzQUNGw4d6SDogw4gz0800" }
  ],

  "shipping_insurance": true, // assicurare la merce
  "shipping_insurance_amount": 1000.0 // valore merce da assicurare
  "shipping_insurance_is_old": false // merce nuova o usata

  "weight": 6.0,
  "price": 7.7,
  "state": "InDelivery",

  "pickup_order": {
    "ticketId": "DF94EQC112929",
    ...
  },
  "notes": [
    { "date": "2016-06-29T10:19:04.1714175", "note": "In spedizione" },
    { "date": "2016-06-29T10:19:04.1485205", "note": "Spedizione prenotata" }
  ],

  "last_update": "2016-06-29T10:19:04.1714175Z",
  "shipping_label_url": "https://...", // url dell'etichetta di trasporto HTML stampabile
  "shipping_label_urls": [ // solo per logistiche avanzate con etichette custom
    { "type": "pdf", "url": "http://.... " },
    { "type": "zpl", "url": "http://.... " },
    { "type": "jpg", "url": "http://.... " }
  ],

  "brt_ddt": "904026", // numero del DDT (da riportare su DDT e ricevuta)
  "brt_label_details": "DATI IN CONFERMA BOLLE DISK A+C\n..." // testo da stampare in evidenza sul DDT
}
```

Appendice A - Etichette, DDT e borderò: formati validi per Fermo!Point

Etichettatura standard per un pacco con servizio SOLO pickup

Le etichette applicate ai pacchi devono contenere alcune informazioni per rendere possibile l'identificazione del pacco sulla consolle di amministrazione del negoziante, così da segnalarne l'arrivo e il ritiro da parte dell'utente senza commettere errori.

L'etichetta può essere generata con il formato richiesto dal corriere, l'importante è che contenga le informazioni qui sotto riportate.

Esistono due varianti da scegliere in base ai dati in possesso di chi crea le etichette, è preferibile utilizzare la prima:

1. Pacco indirizzato direttamente al negozio accompagnato dal logo Fermo!Point e dal Fermo!Ticket. In questo caso il nickname e il **merchantId** NON devono comparire per sicurezza
2. Pacco indirizzato al nickname (accompagnato da **merchantId** se ospite) presso il negozio. In questo caso il Fermo!Ticket NON deve comparire

Esempio formato 1

<p>Destinatario:</p> <p>Cartoleria nuova stella - FP via Roma 47A 20120 Milano MI</p>  <p>Fermo!Ticket: C6D3CBN534742</p>
--

Esempio formato 2

<p>Destinatario:</p> <p>vendonline (20171010001) c/o Cartoleria nuova stella - FP via Roma 47A 20120 Milano MI</p>

NB vendonline (20171010001) è un esempio di nickname ospite creato per l'utente ospite vendonline e il merchantId 20171010001

Etichetta e DDT per il servizio Pickup+Delivery

Nel servizio Pickup+Delivery alla spedizione viene associato un codice (S...) e vengono generate le etichette da applicare ad ogni collo, il DDT da consegnare al corriere e la ricevuta per il merchant in formato HTML. Le etichette vanno stampate ritagliate e utilizzate come indicato direttamente sulle stesse.

Il risultato è lo stesso sia che si prenoti la spedizione tramite nostra consolle, sia integrandosi con le API di spedizione.

Esempio di etichetta + DDT + ricevuta in HTML per il servizio Pickup+Delivery

 <p>Fermo!Ticket: DC443XJ223683</p> <p>Data: 15/05/2017 Numero collo: 1 di 1 Peso del collo in kg: 1 Assicurata da: Fermo!Point Contrassegno: NO</p>	<p>Codice del collo - (da leggere con lettore barcode)</p>  <p>5580FMW695115001</p> 	
	<p>Mittente: Ilmiofornitore Viale Caduri di Nassiriya, 22 70124 - Bari - BA - Italia</p> <p>Contatto: Fermopoint srl 035-690901</p>	<p>Destinatario: La Cartola Via della Ferriera 4 40133 - Bologna - BO - Italia</p> <p>Contatto: Anna Secchieri 051381186</p>

applicare sul pacco 1 di 1 in un punto visibile



 <p>Spedizione: S580FMW695115</p> <p>Data: 15/05/2017 Numero colli: 1 Peso totale in kg: 1 Assicurata da: Fermo!Point Contrassegno: NO</p>	<p>DDT - (indicazione per il corriere)</p> 	
	<p>Mittente: Ilmiofornitore Viale Caduri di Nassiriya, 22 70124 - Bari - BA - Italia</p> <p>Contatto: Fermopoint srl 035-690901</p>	<p>Destinatario: La Cartola Via della Ferriera 4 40133 - Bologna - BO - Italia</p> <p>Contatto: Anna Secchieri 051381186</p>

copia per il corriere



 <p>Spedizione: S580FMW695115</p> <p>Data: 15/05/2017 Numero colli: 1 Peso totale in kg: 1 Assicurata da: Fermo!Point Contrassegno: NO</p>	<p>Copia DDT - (firma/timbro/etichetta del corriere)</p> 	
	<p>Mittente: Ilmiofornitore Viale Caduri di Nassiriya, 22 70124 - Bari - BA - Italia</p> <p>Contatto: Fermopoint srl 035-690901</p>	<p>Destinatario: La Cartola Via della Ferriera 4 40133 - Bologna - BO - Italia</p> <p>Contatto: Anna Secchieri 051381186</p>

copia per il mittente: ricevuta di consegna al corriere



Sistema di automazione tramite Soluzione A+C di BRT

In cosa consiste

L'etichetta BRT presenterà un codice a barre per ogni collo, un DDT (o borderò per più spedizioni) codificato con una dicitura fissa dettata dal corriere. In questo caso il corriere non applicherà etichette adesive ma utilizzerà il codice a barre fornito per l'identificazione. Il tracciamento del pacco può essere fatto tramite l'identificativo di spedizione FIP.

Quando è possibile utilizzare la Soluzione A+C

BRT permette la creazione di un sotto-codice cliente da poter utilizzare con questa soluzione solamente per flussi di pacchi costanti: **almeno 5-10 pacchi ogni giorno, tutti i giorni.**

Formato etichette per BRT Soluzione A+C

Ogni collo sarà identificato da un codice collo univoco (Id spedizione + numero collo su 3 caratteri. Es. S123AD1234553001, S123AD1234553002) che verrà stampato nell'etichetta sotto forma di codice a barre in formato **code128**.

Inoltre è necessario fornire al corriere anche un DDT per ogni spedizione che abbia una dicitura specifica e riporti il codice cliente di BRT, la filiale operativa e il numero di DDT.

I DDT sono raggruppabili in un unico borderò che abbia le informazioni qui sopra per ogni spedizione.

Esempio di etichetta + DDT + ricevuta in HTML da stampare manualmente

 Fermo!Ticket: D50EFVL226422 Data: 31/03/2017 Numero collo: 1 di 1 Peso del collo in kg: 1 Assicurata da: Fermo!Point Contrassegno: NO	Codice del collo - (da leggere con lettore barcode)  S958ACD438772001		
	Mittente: Kaffito Kaffito c/o Asendia Log Solution srl Via Orio Al Serio, 4 24050 - Grassobbio - BG - Italia Contatto: Fermopoint srl 035-690901	Destinatario: Tabacchi -Brescia Via San Polo 230 25134 - Brescia - BS - Italia Contatto: Bocchio Simone 030 2302318	

applicare sul pacco 1 di 1 in un punto visibile



 Spedizione: S958ACD438772 Data: 31/03/2017 Numero colli: 1 Peso totale in kg: 1 Assicurata da: Fermo!Point Contrassegno: NO	DDT - N° 108865 - (indicazione per il corriere) DATI IN CONFERMA BOLLE DISK A+C CODICE CLIENTE 0942606 FERMOPOINT AS400 FIL. 110		
	Mittente: Kaffito Kaffito c/o Asendia Log Solution srl Via Orio Al Serio, 4 24050 - Grassobbio - BG - Italia Contatto: Fermopoint srl 035-690901	Destinatario: Tabacchi -Brescia Via San Polo 230 25134 - Brescia - BS - Italia Contatto: Bocchio Simone 030 2302318	

copia per il corriere



Automatizzare la stampa dell'etichetta

Se il flusso di spedizioni non è gestibile manualmente sarà necessario automatizzare la stampa delle etichette.

Fermo!Point non fornisce un sistema di stampa automatica, ma mette a disposizione del merchant tutti i dati necessari alla creazione dell'etichetta, così da rendere possibile la stampa automatica con il sistema che già utilizza per altri servizi (es. ZPL).

Il formato dell'etichetta dovrà essere verificato da Fermo!Point prima della messa in produzione.

Per effettuare dei test di automazione della stampa su sandbox è necessario utilizzare delle credenziali apposite abilitate al servizio automatico di spedizione BRT Sol A+C:

Client_id:

Lg756a3ZS9As4JKowN2l

Client_secret:

MKAqtYMoBeOp7vsJ3g\$U45lD82qhAjS1#oBsLlJv%18s!58b

Utente Ospite:

guestbrt

2017-02-07

Formato valido per etichette ZPL e BRT Sol A+C

Sull'etichetta dovranno necessariamente essere presenti questi dati:

- Il codice a barre relativo al collo a cui si riferisce l'etichetta (barcode_text)
- Il codice Fermo!Ticket del pickup collegato (D.... o C....)
- La dicitura/logo Fermo!Point
- Indirizzo e contatti del magazzino di ritiro
- L'indirizzo fisico del negozio di destinazione e il contatto in negozio
- Il numero del collo a cui l'etichetta si riferisce e il numero di colli totali
- Il peso del collo a cui si riferisce l'etichetta

Attenzione! In questa modalità NON deve essere stampato il nickname/merchantId del cliente (è già presente il Fermo!Ticket che contiene tutte le informazioni).

Il codice a barre deve essere generato secondo lo standard code 128, la risposta alla chiamata API riporta **per ogni collo** il testo (barcode_text) e l'eventuale stringa in code128 codificata in base64 (barcode_code128_base64).

Il barcode può essere generato da un vostro sistema codici a barre in formato code128 partendo dal il testo (barcode_text), oppure dovrà essere decodificato da base64 il campo barcode_code128_base64 e utilizzato il risultato per scrivere il codice a barre con un font apposito come quello che allego.

Inoltre sarà possibile sostituire i DDT per il corriere in un unico **borderò**, anch'esso validato.

Nel borderò devono essere riportate tutte le informazioni presenti nell'esempio, in particolare la BRT_label_details in alto in evidenza e per ogni spedizione l'id spedizione e il brt_ddt.

A seguito gli esempi di etichetta per un collo e borderò.

Esempio di etichetta da applicare al collo per ZPL




Fermo!Ticket del pickup → **Fermo!Ticket:**
C005EWC806143

Data: 09/02/2017

Collo: 1 di 2
numero collo e totale

Peso collo in kg: 1.5
peso collo

Assicurata: NO
Contrassegno: NO

id ordine: test 1, **id uds:** 00196919 ← **Eventuali note**

barcode_code128 → **barcode_text** → **SABEDQL199479001**

Mittente: Quickly Logistics House S.r.l. via Roggia Vignola, 9 24047 - Treviglio - BG	Destinatario: Cafè Colombo Piazza Roma, 7 22010 - Argegno - CO
Contatto: Mario rossi 0331 1850316	Contatto: Andrea 031821124

↑ **Magazzino di ritiro e contatto mittente**
↑ **Point di destinazione Contato negoziante**

Esempio di borderò per ZPL

BRT_label_details ↓

DATI IN CONFERMA BOLLE DISK A+C CODICE CLIENTE 0942606 FERMOPOINT AS400 FIL. 222

Mittente:
Nome merchant
indirizzo magarazzino di ritiro merchant

deliveryId / brt_ddt ↓

ORDINE nr.	DESTINATARIO	INDIRIZZO	ID SPEDIZIONE / DDT n.	PESO (kg)	COLLI
840974	mario rossi	Cartoleria nuova stella - FP via roma, 10 20120 Milano MI Contatto: 02 23447563	S55CDMP42921 / 001285	2,1	2
840975	luca bianchi	Ortopedia Lupi - FP via roma, 10 20120 Milano MI Contatto: 02 23447563	S54DDAP40114 / 001286	2,1	2
840976	filippo bosnaghi	Edicola Master - FP via firenze, 10 20120 Milano MI Contatto: 02 23447563	S33HGRP404837 / 001287	2,1	2
840980	lorenzo medici	Cartoleria nuova stella - FP via roma, 10 20120 Milano MI Contatto: 02 23447563	S852DRRE12736 / 001285	2,1	2

TOTALE SPEDIZIONI: 1

TOTALE PESO: 2,1

TOTALE COLLI: 2

DATA: _____

FIRMA AUTISTA: _____