

Webshop E-commerce Integráció

Foxpost, info@foxpost.hu

January 8, 2015 (v1.0r4)

Bevezetés

Ez a leírás bemutatja, hogy a webes E-commerce szolgáltatások hogyan használják majd a csomag szállítási hálózatot. A felület három fő esetet támogat, melyek szükségesek az Ecommerce rendszerekhez.

1. Biztosít egy listát a vásárlóknak a lehetséges átvételi pontokról.
2. Létrehoz egy szállítási megrendelést az átvételi pontokhoz a vásárlóknak.
3. Csomag követési lehetőség

A felület lazán követi a REST[2] elveit és a biztonságos kommunikációhoz HTTP protokollt TLS[1]-t használ. Az egész felület UTF-8 karakterkódolást és JSON, illetve XML adatformátumot fogad el. REST API url: <https://api.foxpost.hu/ecom/>

1. Források

A következő lista tartalmazza azon forrásokat amelyek az API-n keresztül elérhetőek. Ezen forrásokat tudják az E-commerce rendszerek lekérdezni, illetve frissíteni.

- places --- Lehetséges Terminálok
- orders --- Megrendelések
- traces --- Csomagkövetési információk □ Languages --- Támogatott nyelvek.

Minden egyes erőforrás meghatározott számú megnevezett mezőből áll, amelyek az erőforrások bemutatásoknál vannak pontosan definiálva. A mezők tulajdonságai szögletes zárójelben van megadva a mező neve után. A típus információk tartalmazzák az adott karakterlánc maximális megengedett hosszát, illetve számok esetében, hogy hány tizedes jegyet engedélyez.

1.1 Terminálok

Az erőforrások megjelenítése és ezen erőforrások támogatott kérései a következő alpontokban vannak részletezve.

A következő paramétereket (csak olvasható) a lokációk lekérésénél és listázásnál kaphatja meg. Az E-commerce rendszereknek a lokációkat frissíteniük kell.

- place_id [integer] --- Egyedi azonosítója a terminálnak. Ez határozza meg, hogy melyik terminálba történt a megrendelés.
- name [string(70)] --- A terminál neve
- address [string(100)] --- A terminál címe
- group [string(35)] --- A group azaz csoport tartalmazza a régiót vagy a megyét (várost)
- subgroup [string(35)] --- A group azaz csoport tartalmazza a várost
- open [string(75)] --- A lokáció nyitva tartása
- findme [string(254)] --- Segítség a terminál megtalálásához
- geolat [decimal(8,6)] --- GPS Szélességi koordináták
- geolng [decimal(8,6)] --- GPS Hosszúsági koordináták

A mező név, open és findme lokalizáltak. Részletesebb leírás az 1.5-ös pontban talál. A place_id, és a name mezőknek mindig van értéke, de a többi mező lehet üres. a group és a subgroup mezők segítséget nyújtanak egy felhasználóbarát és hierarchikus lista elkészítéséhez.

1.1.1

Az 1.1 és az 1.2 es példa szemlélteti, hogy milyen formátumban és milyen adatokat kap vissza a felhasználó, amikor lekéri a terminál listát.

1.2 Rendelések

Ez az a paraméter, amelyet új megrendelés készítésénél használunk, hogy a meglévő rendelésekről információt kapjunk. A rendelések forrás Több módot használ. A POST új megrendelések létrehozásakor, A GET a meglévő megrendelések listázáshoz és a rendelési adatok felbontásához, a PATCH és PUT a frissítéshez és az áthelyezéshez, és a DELETE pedig a meglévő megrendelés törléséhez.

A következő lista tartalmazza a megrendelés mezőit. Figyelembe véve az állandó, a lehetséges és csak az olvasható mezőket. Az állandó mezők értékeit meg kell adni POST kérelemnél, amely az új megrendelés létrehozásához szükséges. Azon mezők értéke, amelyek csak olvashatók automatikusan a rendszer által generált, és ha ezeket az értékeket megváltoztatjuk, annak az eredménye hibát okoz.

```

{
  "place_id":1001,
  "name":"Extra Nice Shopping Centre",
  "address":"Street 123, City",
  "open":"Mon-Fri from 9am to 8pm"
},
{
  "place_id":2007,
  "name":"The Gas Station",
  "address":"1001 Another Street, Town",
  "open":"From 9am to 8pm, except holidays"
}, ...
]

```

Figure 1.1: Terminál lista JSON formátumban

```

<places>
  <place href="https://host/path/places/1001">
    <place_id>1001</place_id>
    <name>Extra Nice Shopping Centre</name>
    <address>Street 123, City</address>
    <open>Mon-Fri from 9am to 8pm</open>
  </place>
  <place href="https://host/path/places/2007">
    <place_id>2007</place_id>
    <name>The Gas Station</name>
    <address>1001 Another Street, Town</address>
    <open>From 9am to 8pm, except holidays</open>
  </place> ...
</places>

```

Terminál lista XML formátumban

- barcode [string(100)] --- A barcode egy rendszer által generált érték amelyet a rendelések követéséhez használ.
- barcode_alt [string(100)] --- Egy lehetséges külső vonalkód amellyel a csomagokat lehet ellátni.
- place_id [integer] --- a cél terminál azonosítója (állandó, nem üres).
- name [string(254)] --- A vásárló neve □ phone [string(50)] --- A vásárló telefonszáma □ email[string(254)] --- A vásárló e-mail címe:
- content [string(254)] --- Lehetséges leírás
- comment [string(1000)] --- megjegyzés
- refcode [string(254)] --- lehetséges hivatkozás ehhez a megrendeléshez amellyel le lehet kérni a rendelés adatait és a hozzá lehet kapcsolni a megrendelést a vevőhöz.
- language [string(25)] --- Az a nyelv amelyen a vásárlóval lehet kommunikálni.
- cod_amount [integer] --- Utánvétkori fizetés a vásárlónak akkor kell fizetnie mielőtt átvénné a csomagot.

– EUR 123.45 → cod amount = 12345 cod currency = EUR

– CLP 12345 → cod amount = 12345 cod currency = CLP

Az érték [0,2e9] lehet. Az érték a megrendelés leadása után már nem módosítható.

- cod_currency [string(3)] — ISO 4217 [5] Valuta kód (állandó amikor a cod_amount értéke megvan adva)
- timestamp [string] --- A rendelés feladásának pontos ideje RFC3339 [6] formátum (csak olvasható)
- status [string] --- a megrendelés státusza az 1.1-es táblázatban meghatározott értékeket veheti fel. (csak olvasható).
- tracc [string] --- Hivatkozása rendelés követéshez.

1.3 Példák

Egy megrendelés létrehozásához az ORDERS erőforrásnak kell elküldeni az adatokat POST használatával. Az 1.3-as példa mutatja a szükséges adatokat. Sikeres rendelés leadása esetén az 1.4-es példában láthatjuk a visszatérő értéket. A JSON formátumú példákat az 1.5 és az 1.6-os pontban mutatjuk be.

Státusz	Leírás
New	Új megrendelés
to-term	A csomag úton van a csomagelosztóba
in-term	A csomag a csomagelosztóban van
to-apt	A csomag a terminálhoz tart
in-apt	A csomag a terminálban van
delivered	A csomag kézbesítve lett a vásárlónak
ret-in-apt	Visszaküldött csomag a terminálban
ret-to-term	A csomag úton van vissza a csomagelosztóba
ret-in-term	A visszaküldött csomag a csomagelosztóban
ret-to-apt	A visszaküldött csomag úton a terminálhoz
lost	A csomag elveszett
cancelled	A megrendelés törölték

1.1 Tábla a rendelési állapotokról

```

<order>
  <place_id>2007</place_id>
  <name>John Smith</name>
  <phone>+123451234</phone>
  <email>john@smith.com</email>
</order>

```

1.3: Új megrendelés leadása XML formátumban

```

<order href="https://host/path/orders/EE123456789">
  <barcode>EE123456789</barcode>
  <place_id>2007</place_id>

```

```
<name>John Smith</name>
<phone>+123451234</phone>
<email>john@smith.com</email>
<language>en-GB</language>
<timestamp>2012-06-21T08:50:08+03:00</timestamp>
<status>new</status>
<trace href="https://host/path/traces/EE123456789"/>
</order>
```

1.4: XML formátumban a válasz

```
{
  "place_id": 2007,
  "name": "John Smith",
  "phone": "+123451234",
  "email": "john@smith.com"
}
```

1.5: Új megrendelés leadása JSON formátumban

```
{
  "barcode": "EE123456789",
  "place_id": 2007,
  "name": "John Smith",
  "phone": "+123451234",
  "email": "john@smith.com",
  "language": "en-GB",
  "timestamp": "2012-06-21T08:50:08+03:00",
  "status": "new", "trace": { "href": "https://host/path/traces/EE123456789" }
}
```

1.6: JSON formátumban a válasz

1.4 Rendelés követés

Ezt a csak olvasható erőforrást a csomagok követésére és a szállítás státuszára használhatjuk. A csomagkövetés időbélyeggel ellátott eseményeken alapszik. Minden egyes esemény tartalmazza a „NEW” státuszt, adott esetben a helyet és az eseményhez, hogy hol fogadható el.

- type [string(35)] --- Az esemény típusa (létrehoz, töröl, szállítás)
- timestamp [string] --- Pontos idő az eseményhez: RFC 3339 [6] format
- status [string(35)] --- A szállítás állapota az esemény után lásd még 1.2 és 1.1 táblák
- location [string(100)] --- A hely ahonnan az eseményt kezdeményezték □ initiator [string(100)] --- Az esemény kezdeményezője

Bármiféle előre definiált erőforrás nélkül a követés erőforrás egy listát hoz létre az elérhető követésekről. A rendelés vonalkódját (barcode) kötelező megadni, mint egy erőforrást, ahhoz hogy rendelés adatait lehessen listázni. A kérés elküldése egy rendelés adatainak megtekintéséhez a következő formátumot kell használni:

<https://host/path/traces/EE123456789>.

1.4.1 példák

A GET kérés formátuma a rendelés adatainak lekéréshez az 1.7 és az 1.8 as példában található. A követés adatai megvizsgálhatók a következő linkek által, illetve ha valós barcode van megadva az URL-ben 1.9 és 1.10 es példa.

1.5 Nyelvi erőforrások

Ez a csak olvasható erőforrás tartalmazza a támogatott nyelveket (tags). A nyelvi tag két részből áll mely kötőjellel van elválasztva.

```
<traces>
```

```
<trace barcode="EE123456789" href="https://host/path/traces/EE123456789"/>
```

```
<trace barcode="ES1234567890" href="https://host/path/traces/ES1234567890"/>
```

```
...
```

```
</traces>
```

1.7: Követések XML formátumban

```
[  
  {"href": "https://host/path/traces/EE123456789",  
    "barcode": "EE123456789" },  
  {"href": "https://host/path/traces/ES1234567890",  
    "barcode": "ES1234567890" },  
  ...  
]
```

1.8: Követések JSON formátumban

```
<trace barcode="EE123456789">  
  <event type="created" timestamp="2012-06-21T08:50:08+03:00" status="new"/>  
  <event type="cancelled" timestamp="2012-06-21T09:50:08+03:00" status="cancelled" initiator="Customer  
X"/>  
</trace>
```

1.9: Követés XML ábrázolása

```
[  
  {"type": "created",  
    "timestamp": "2012-06-21T08:50:08+03:00",  
    "status": "new" },  
  {"type": "cancelled",  
    "timestamp": "2012-06-21T09:50:08+03:00",  
    "status": "cancelled",  
    "initiator": "Customer X" }  
]
```

1.10: A követés JSON ábrázolása

Az első két betű a nyelvi kód az ISO 639-1 [4] szabvány alapján. A második rész pedig az ország kód az ISO 3166 [3] szabvány alapján. A következő példa lista megmutatja a nyelvi tagokat és a hozzá tartozó nyelvet, amely a GET kérelem alapján következő.

- es-ES ---- Spanyol
- eu-ES ---- Baszk
- ca-ES ---- Katalán
- en-GB ---- Brit Angol

A nyelvi tagok a felület nyelvét határozzák meg. Amikor a kliens egy úgynevezett lokalizált választ vár akkor az Accep-Header paramétert ennek megfelelően kell beállítani. Amikor nincsen beállítva a fentebb említett header értéke akkor az alapértelmezettre beállított nyelv értékét fogja felvenni. A szerver magába foglalja a Content-Language értékét válaszkor. Ha nem támogatott nyelv hívódik meg akkor a szerver 406-os válaszkódot állít be (nem elfogadható) Nyelv nem támogatott.

1.5.1 Példák

A nyelvi erőforrás példáját az 1.11 es és az 1.12 es részben van megadva. Habár a nyelvi tag kevert betűkkel van megadva (kis és nagybetű), ugyanúgy kell megadni a http fejlécben is.

2. Szállítási Biztonság

Az adat kommunikáció egy biztonságos kódolt csatornán történik (HTTPS protokoll). A klienseknek el kell fogadniuk a tanúsítványt minden egyes folyamat előtt. A szerver címét az igazolást és az újlenyomatot külön-külön tartalmazza a dokumentum.

3. Azonosítás

Minden kérést, folyamatot azonosítani kell. Az egyedüli elérhető azonosítási mechanizmus az a HTTP alap azonosítás, amely széles körben támogatott.

```
<languages>
```

```
<language default="true" tag="es-ES"/>
```

```
<language tag="eu-ES"/>
```

```
<language tag="ca-ES"/>
```

```
<language tag="en-GB"/>
</languages>
```

1.11 Nyelv XML formátum

```
[
  { "tag": "es-ES",
    "default": "true" },
  { "tag": "eu-ES" },
  { "tag": "ca-ES" },
  { "tag": "en-GB" }
]
```

1.12 Nyelv JSON formátum

HTTP Klienseket és a HTTP library-kat, általában kliens oldalon történnek. Példa az azonosításra cURL használatával.

```
curl -u username:password https://host/path/
```

Protokoll szinten, minden kérés magában foglalja az Authorization header-t. A header-nek tartalmaznia kell a Basic X értéket ahol az X a Base64 titkosításból származó username:password. Például ha a username az username és a password meg password akkor a következő képen néz ki.

```
Authorization: Basic dXNlcm5hbWU6cGFzc3dvcmQ=
```

4. Kérések és Formátumok

Minden egyes erőforrás Különböző formátumokat képvisel, mely a jelenlegi specifikáció alapján JSON és XML. Ez ugyancsak elfogadott a kliens által küldött kérésekben. Ha a kliens XML formátumban küld adatot a szerver válaszol egy másik formátumban például JSON.

A formátumok a szabványos HTTP headers accept és content-type.

- application/vnd.cleveron+json; version=1.0
- application/vnd.cleveron+xml; version=1.0

Ha nincs definiálva az accept és a content-type akkor az alapértelmezett adatformátum a JSON. A verzió szám támogatja a jövőbeli fejlesztéseket és a visszafele kompatibilitást is. A verzió szám nem növekszik a kisebb vagy a visszafele kompatibilis változásoknál. Ha nincs verziószám specifikálva, akkor a legutolsó támogatott verzió lép érvénybe. A jelenlegi verzió nem tartalmazza a „HTTP media range”-eket. Például ha a kliens XML formátumban küld kérést és JSON-ban várja vissza az választ, akkor a Kliensnek be kell állítani a következő paramétereket.

Accept: application/vnd.cleveron+json

Content-Type: application/vnd.cleveron+xml

Ahogy a JSON az alapértelmezett formátum az első header elhagyható ebben az esetben. Ezen kívül hogy sávszélességet nyerjünk illetve csökkentjük a válasz időt a kliens kérhetnek tömörítést a fejlécben azáltal, hogy a header-t a következő módon állítják be:

Accept-Encoding: gzip, deflate

Ebben az esetben a szerver tömörített adatot küld vissza.

5. Erőforrás módok

Öt darab HTTP metódus áll rendelkezésünkre GET, POST, PUT, PATCH and DELETE. A következőkben a módok szemantikáját mutatjuk be.

5.1 GET

Get metódust az adatok lekérésére használjuk például:

```
curl -X GET -u username:password \  
-H "Accept: application/vnd.cleveron+xml; version=1.0" \  
-H "Content-Type: application/vnd.cleveron+xml; version=1.0" \  
https://host/path/
```

A GET metódus használatakor garantált, hogy semmilyen erőforrás nem változik, a kérés megismételhető és a válasz esetleg bekerülhet a cache kliens vagy proxy által.

5.2 POST

A Post metódussal új erőforrást hozhatunk létre. Az egyetlen erőforrás amit POST-on keresztül át tudunk adni az az Orders. A válasz tartalmazza a létrehozott erőforrást.

5.3 PUT

A PUT metódus egy már meglévő erőforrás felülírására használható. Ez a metódus az order erőforráson alkalmazható addig, amíg a cél erőforrás állapota NEW. A válasz tartalmazza új erőforrások ábrázolását.

5.4 DELETE

A DELETE metódust a rendelések törlésére használható. A rendelés nem törölhető miután kiszállították a megrendelőnek. A válasz törzse üres. A törlés nem visszavonható. A törlést nem lehet sikertelen.

5.5 PATCH

A PATCH metódus módosítja egy erőforrás meglévő mezőjét. Csak az orders erőforrásokon alkalmazható. Ez a metódus használható, hogy átirányítsanak egy még nem kézbesített csomagot, vagy megadni egy külső barcode/label értéket.

6. Responses

Minden választ tartalmaz egy HTTP válasz kód. A következő lista bemutatja a lehetséges kódokat és értelmezésüket.

- 200: Siker (Sikeres GET, PUT vagy DELETE kérésnél)
- 201: Elkészült (Sikeres POST kérés esetén)
- 400: érvénytelen erőforrás (Rosszul formázott kérés)
- 401: Azonosítási hiba (Hibás vagy hiányzó adatok)
- 404: Erőforrás nem található

- 405: Nem engedélyezett metódus
- 406: Nem elfogadható érték / paraméter / tag / subscription
- 500: Alkalmazás hiba

Sikeres POST kérésnél a visszatérő státuskód az 201, magába foglalva a location headert mely tartalmazza az elkészült erőforrás URL-jét, és törzs tartalmazza az elkészült erőforrást. Sikeres PUT kérés esetén a visszatérő kód az 200, és a törzs tartalmazza a frissített erőforrást. Sikeres Törlés esetén A visszatérő kód 200 és üres törzzsel tér vissza.

7. Áttekintés

A 7.1 es tábla tartalmazza a verzió történetet és a jelentős változásokat.

Version Date Description

1.0r2	May 23, 2014	Initial public version	
1.0r3	November 12, 2014	Added geolat, geolng to places; fixed typos	1.0r4
	January 8, 2015	Added cod amount, cod currency	

Table 7.1: Áttekintés

References

[1] T. Dierks and E. Rescorla. The Transport Layer Security (TLS) Protocol Version 1.2. RFC 5246, August 2008. Updated by RFCs 5746, 5878, 6176.

[2] Roy T. Fielding and Richard N. Taylor. Principled design of the modern web architecture. ACM Trans. Internet Technol., 2(2):115–150, May 2002.

- [3] International Organization for Standardization. ISO 3166-1:2006 Codes for the representation of names of countries and their subdivisions — Part 1: Country codes. ISO, Geneva, 2 edition, 2006.
- [4] International Organization for Standardization. ISO 639-1:2002 Codes for the representation of names of languages — Part 1: Alpha-2 code, 2002.
- [5] International Organization for Standardization. ISO 4217:2008 – International Standard for currency codes, 2008.
- [6] G. Klyne and C. Newman. Date and Time on the Internet: Timestamps. RFC 3339 (Proposed Standard), July 2002.