



PUDO Partner Interfész specifikáció

Készítette:

SPRINTER Futárszolgálat Kft.

Budapest, 2021.



Dokumentum történet

Dátum	Módosították	Változások	Verzió
2017.02.23	Kovács Richárd	Létrehozás	1.0
2017.02.28	Szűcs Krisztián	Véglegesítés	1.1
2017.06.21	Kovács Richárd	Token mező	1.2
2017.10.27	Kovács Richárd	Okmány visszaforgatás	1.3
2018.03.22	Szűcs Krisztián	Tracking	1.4
2018.07.13	Szűcs Krisztián	Csomag értékek	1.5
2019.07.29	Szűcs Krisztián	Címke előre nyomtatás, A6	1.6
2020.11.16	Tóth Dávid	ZPL	1.7
2021.02.01	Zagyi Péter	Háromszereplős csomagfeladás	1.8



Tartalomjegyzék

1. BEVEZETŐ	5
2. KOMMUNIKÁCIÓ	5
2.1. ÁLTALÁNOS INFORMÁCIÓK	5
3. CSOMAGREGISZTRÁCIÓ	6
3.1. ÁLTALÁNOSAN A CSOMAGREGISZTRÁCIÓRÓL	6
3.2. PICK PACK PONT	7
3.2.1. NORMAL – RAKTÁRBÓL FELADOTT CSOMAG PPP-RE	7
3.2.2. POS2POS – PPP FELADATOTT CSOMAG MÁSIK PPP-RE	7
3.2.3. DIREKT – PPP-N FELADOTT ÉS ÁTVETT CSOMAG	8
3.2.4. RETURN – PPP-N FELADOTT VISSZÁRUS CSOMAG A PARTNERNEK	8
3.3. HÁZHOZ SZÁLLÍTÁS	8
3.3.1. NEMZETKÖZI HÁZHOZSZÁLLÍTÁS - FANCOURIER	8
3.4. VONALKÓD CSERE	9
3.5. OKMÁNY VISSZAFORGATÁS	9
3.5.1. PICK PACK PONT	9
3.5.2. HÁZHOZ SZÁLLÍTÁS	10
3.6. TÍPUSOK, MEZŐK LEÍRÁSA	10
3.6.1. Kiemelt fontosságú mezők	10
3.6.1.1. SERVICETYPE	10
3.6.1.2. TRANSITTIME	10
3.6.1.3. PARCELCOUNT	10
3.6.1.4. RETURNOFDOCUMENT	11
3.6.2. FÜGGVÉNYEK	11
REGISTERPARCELCONTAINER	11
3.6.3. TÍPUSOK	11
3.6.3.1. REQUEST	11
REGISTERPARCELCONTAINERREQUEST	11
PARCEL	12
SUPPLIER	15
CUSTOMERTYPE	15
PARCELDeliveryType	16
PACKAGEType	16
PARCELcreationStatus	16
PARCELSERVICEType	17
3.6.3.2. RESPONSE	17
REGISTERPARCELCONTAINERRESPONSE	18
PARCELRESULT	18
3.6.3.3. HIBAKÓDOK	19
ERRORCODE	19
4. CÍMKE NYOMTATÁS	22
4.1. ÁLTALÁNOSAN A CÍMKEGENERÁLÁSRÓL	22
4.2. CÍMKE NYOMTATÁS CSOMAG REGISZTRÁCIÓ NÉLKÜL	22
4.3. Milyen dokumentumokat kaphat vissza	23
4.3.1. CÍMKE/FUVARLEVÉL	23
4.3.1.1. 2X2-ES MÉRET – A4-ES LAP 4 RÉSZRE BONTVA	23
4.3.1.2. A6-OS MÉRET (HŐPAPÍROS- ÉS SZALAGNYOMTATÓHOZ)	24
4.3.1.3. 2X4-ES MÉRET – A4-ES LAP 8 RÉSZRE BONTVA	24



4.3.1.4. NEMZETKÖZI FANCOURIER CSOMAG CÍMKE	25
4.3.2. CSOMAGÁTVÉTELI ELISMERVÉNY	25
4.3.2.1. FIXPONTOS SZÁLLÍTÁS – PPP	26
4.3.2.2. HÁZHOZ SZÁLLÍTÁS – HD	26
4.3.3. KÉSZPÉNZÁTVÉTELI ELISMERVÉNY	27
4.4. TÍPUSOK, MEZŐK LEÍRÁSA.....	29
4.4.1. FÜGGVÉNYEK.....	29
GETDOCUMENT	29
4.4.2. TÍPUSOK	29
4.4.2.1. REQUEST	29
MASSDOCUMENTREQUEST.....	29
BARCODEDATA.....	29
BARCODETYPE.....	30
DOCUMENTSETTING	30
DOCUMENTFORMAT	30
LABELDOCUMENTPOSITION	30
LABELDOCUMENTSIZE	31
DOCUMENTTYPE.....	31
4.4.2.2. RESPONSE.....	32
MASSDOCUMENTRESPONSE	32
DOCUMENTRESULTS	32
DOCUMENTGENERATIONRESULT.....	32
RESULTCODE.....	33
DOCUMENTDATA	33
4.4.2.3. HIBAKÓDOK.....	34
5. FORRÁSKÓD DEMÓ	34
5.1. C# KÓD.....	34
5.1.1. CSOMAGREGISZTRÁCIÓ	34
5.1.1.1. HIBAKEZELÉS ÉS ADATFELDOLGOZÁS	36
5.1.2. CÍMKE GENERÁLÁS.....	37
5.1.2.1. HIBAKEZELÉS ÉS ADAT FELDOLGOZÁS	40
5.2. PHP KÓD	41
5.2.1. CSOMAGREGISZTRÁCIÓ	41
5.2.1.1. HIBAKEZELÉS ÉS ADATFELDOLGOZÁS	43
5.2.2. CÍMKE GENERÁLÁS.....	44
5.2.2.1. HIBAKEZELÉS ÉS ADATFELDOLGOZÁS	46
6. CSOMAG NYOMKÖVETÉS (TRACKING).....	48
6.1. REQUEST	48
UNIFIEDPARCELTRACKINGINFO.....	48
TRACKINGINFO	49
FŐBB MEZŐK.....	49
STÁTUSZOK	49
EGYÉB AZONOSÍTÓK.....	50
EGYÉB INFORMÁCIÓK.....	51
TRANSLATION.....	52
6.2. RESPONSE	52
6.2.1. HIBAKÓDOK.....	52
6.3. FORRÁSKÓD DEMÓ.....	53

6.3.1. C# KÓD	53
6.3.2. PHP KÓD.....	53
6.3.3. TESZTELÉS.....	54

1. Bevezető

Az interfész dokumentum célja, hogy részletezze, és példákon keresztül bemutassa, miként lehet a PUDO rendszerbe csomagokat rögzíteni mind Pick Pack Pontra, mind házhozszállításra külső alkalmazásból. Továbbá, hogy hogyan lehet a beregisztrált csomagokról a PUDO rendszer dokumentumgeneráló funkciójának segítségével - szintén külső alkalmazásból - lekérni a címkéket. A dokumentum részletesen tárgyalja a funkciók elérésének technikai megvalósítását és üzleti szabályait is, melyeket a dokumentumban 1-1 példa kóddal is szemléltetünk. A példa kódok .Net C#, valamint PHP nyelven kerülnek bemutatásra. A példa kódok felépítésénél figyelembe vettük az adatszerkezetek fokozatos felépítésének szabályait.

2. Kommunikáció

A kommunikáció HTTP protokollon keresztül történik, HTTP 1.1 specifikáció szerinti üzenetekkel, UTF-8 kódolással. Egy tranzakciót a kliens egy HTTP request küldésével indít, melyre a válasz a HTTP response-ban megy vissza. Ez a kommunikáció WCF service-en keresztül valósul meg.

2.1. Általános információk

A paraméterek típusa XSD (XML Schema Definition Language, <http://www.w3.org/TR/xmlschema-0/>) szerinti típust jelent.

Néhány adattípus magyarázata:

int

- Értéke: 4 byte-os előjeles érték (+2,147,483,647-től -2,147,483,648-ig)
- Reprezentációja: számok karakterlánc (pl. „14400”)

boolean

- Értéke: igaz/hamis
- Reprezentációja: „true” és „false”, vagy „1” és „0”

dateTime

- Értéke: érvényes dátum i.sz. 1900-2100-ig
- Reprezentációja: „CCYY-MM-DDThh:mm:ss”, ahol a 'CC' az évszázadot, az 'YY' az évet, a 'MM' a hónapot, a 'DD' a napot, a 'hh' az órát, az 'mm' a percet, az 'ss' pedig a másodpercet jelöli 0-val balról feltöltve. A '-', 'T' és a ':' elválasztójelek kötelezőek. (pl. 2003-05-05T16:41:57)

Ha egy metódushívás során hiba történik, akkor az arra adott válasz formája függ attól, hogy a hiba milyen szinten történt:

- A HTTP szintű hibákra HTTP hibaüzenet a válasz. (pl. Not Authorized)



- A SOAP szintű hibákra SOAP Fault üzenet a válasz (pl. dátum paraméter XML reprezentációja helytelen)
- A Pudo alkalmazásszintű hibákra ErrorCode a válasz (pl. hibás partner).

3. Csomagregisztráció

3.1. Általánosan a csomagregisztrációról

Az alább bemutatandó Service hívás az egy partnertől történő csomagregisztrációkat hivatott bemutatni.

A csomagrögzítés során lehet vegyesen is feladni csomagokat, ami azt jelenti, hogy lehet Pick Pack Pontos és / vagy Házhozszállítós csomagokat vegyesen regisztrálni, nem szükséges minden csomag típusnál új Service hívást indítani.

Az egy service hívással beküldött csomagok egy partneri szállítmányba fognak kerülni típusonként!

Ez azért fontos, mivel a később tárgyalt dokumentum, illetve csomagcímke nyomtatásnál is eltérő dokumentumokat kaphatunk vissza a válaszüzenetben, regisztráció után. Az azonos helyről feladott és / vagy azonos helyre érkező csomagok azonos szállítmányra kerülhetnek.

A rendszer egy kérésen belül, maximum 1000 csomag adatot tud lekezelni. Ennél több csomag adatot, biztonsági okokból nem dolgozunk fel és PSR_TOO_MANY_PARCEL_IN_REQUEST hibaüzenetet küldünk vissza, részleges csomagrögzítés ebben az esetben nem lehetséges. **Házhoz szállítós** csomagok esetében egy csomag adaton belül lehetőség van több al-csomagot regisztrálni (ParcelCount mező, ezeknek minden adata azonos), ezek maximális száma 100. Ennek túllépése esetén is PSR_TOO_MANY_PARCEL_IN_REQUEST csomag szintű hiba üzenetet állítunk be. A kérésben szereplő többi csomag regisztrálását ez nem befolyásolja. **Pick Pack Pontos** csomagok esetében ezt a mezőt nem kell töltetni, mivel ilyen feladások esetében maximum 1 darab csomag adatot dolgozunk fel. Egy kérésen belül maximum 1000 db Pick Pack Pontos csomagot lehet feladni. Így a leírtak alapján több csomag regisztrálása interface-en keresztül nem lehetséges.

Csomagonként a beküldött vonalkódok maximum hossza 25 karakter lehet.

Minden Partner egyedi azonosítóval rendelkezik. Így biztosítva, hogy minden csomagfeladás köthető legyen partnerhez.

AZ EGYEDI AZONOSÍTÓ (PREFIX) PARTNERENKÉNT ELTÉR. AZ ELŐTAGOT A SPRINTER IT SUPPORT-TÓL (pudointerface@lapker.hu) KELL IGÉNYELNI!

Ez a kód kerül rá a csomag címkékre is.

TOVÁBBI BIZTONSÁGI MEZŐ, EGY ÚGYNEVEZETT TOKEN (10 KARAKTERHOSSZÚ, KISBETŰKET ÉS SZÁMOKAT TARTALMAZ VEGYESEN) MELYET SZINTÉN AZ IT SUPPORT-TÓL KELL IGÉNYELNI! EZ CSAK A PARTNER SZÁMÁRA KIADHATÓ, ÉS SUPPORT-ON KERESZTÜL IGÉNYELHETŐ A MÓDOSÍTÁSA.



Ennek a mezőnek az értékét kötelezően küldenie kell a partnernek. Amennyiben rossz **Token** lett beküldve, vagy nem lett beküldve a csomagregisztrációval egy időben, úgy a csomag feladása érvénytelen. **PSR_INVALID_PARTNER_TOKEN** csomagszintű hibaüzenetet küld vissza a PUDO minden egy request-ben szereplő csomagra.

A csomagregisztráció sikerességéről válasz üzenetben kapunk egy **RegisterParcelContainerResponse** visszajelzést (melyben az **ErrorCode** mező ad tájékoztatást a regisztráció sikerességére vonatkozóan), valamint egy **ParcelResult** listát, melyben csomagonkénti lebontásban találhatjuk meg a csomagokhoz tartozó adatokat.

Amennyiben sikeres, vagy részlegesen sikeres a csomagok regisztrálása, úgy a PSR_OK választ kapjuk az **ErrorCode** mezőben. A részlegesen sikertelen csomag rögzítéseket, a csomagszintű ErrorCode-ban jelezzük. A további hibakódokat később tárgyaljuk.

3.2. Pick Pack Pont

A Pick Pack Pontos (továbbiakban PPP) csomagregisztrációknál 4 esetet különíthetünk el.

A programkódban egyszerű paraméterezéssel lehet eldönteni, illetve elkülöníteni, hogy melyik típus szerint szeretnénk rögzíteni a csomagjainkat. Ezek a beállítások később kerülnek részletezésre.

PickupLocationId és **DestinationLocationID**-kkel lehet meghatározni a csomagok feladási illetve célba érkezésének helyét. Ezek a LocationID-k PPP-s boltokat azonosítanak, a hívásnál a boltazonosítót kell megadni. Az adatok elérhetőek például a térképes boltkeresőn (beágyazható iframe): <http://online.sprinter.hu/terkep/#/>

3.2.1. Normal – Raktárból feladott csomag PPP-re

Ez a PPP feladási mód azt jelenti, hogy a csomagfeladás helyszíne a partner alapértelmezett feladási raktára és a kézbesítési cél egy PPP-os bolt. Ebben az esetben csak a **DestinationLocationId**-t kell meghatározni.

3.2.2. POS2POS – PPP feladatott csomag másik PPP-re

Ez a PPP feladási mód azt jelenti, hogy egy PPP-n feladott csomagot egy másik PPP-re kell eljuttatni. Ebben az esetben **PickupLocationId** (csomagfelvétel helye) és **DestinationLocationID** (csomagátvétel helye) megadása kötelező.

AMENNYIBEN OLYAN POS2POS CSOMAGOT RÖGZÍTÜNK, AMELYNÉL A PICKUPLOCATIONID ÉS DESTINATIONLOCATIONID UGYAN AZT A SHOPID-T TARTALMAZZA, VAGYIS UGYAN ARRA A BOLTRA MUTAT, AKKOR A DIREKT SERVICE-TYPE-OT VALÓSÍTOTTUK MEG. ILYEN ESETBEN A DIREKT BEÁLLÍTÁST KELL ALKALMAZNI.

3.2.3. Direkt – PPP-n feladott és átvett csomag

Ez a PPP feladási mód azt jelenti, hogy egy PPP-n feladott csomagot ugyanazon a PPP-n fognak átvenni. Vagyis a feladó adott PPP-n a csomagot leadja, és az fizikailag addig marad ott, amíg át nem veszik. Ebben az esetben elegendő, illetve kötelező a **DestinationLocationID** megadása (**PickupLocationId**-t nem kell kitölteni).

3.2.4. Return – PPP-n feladott visszárus csomag a partnernek

Ez a PPP feladási mód azt jelenti, hogy a csomagot vissza kell juttatni a feladónak. Vagyis a vevő PPP-n feladja a csomagot, majd a partner a beszállítási raktárában átveheti. Ebben az esetben szintén a **DestinationLocationID**-t kell megadni.

3.3. Házhoz szállítás

A Házhoz szállítós (később HD) csomag regisztrációknál **ServiceType** szerint 1 eset létezik, viszont ezen belül eltérhet a szállítás módja (**DeliveryType**).

- Lehet Kiszállítás (**OnlyDelivery**), ilyenkor a partner adja fel a csomagot (szerződés szerint érte meg a futár telephelyre, vagy behozza raktárba) és kiszállításra kerül a vevő egyedi címére.
- Lehet Lehívás (**OnlyReturns**), amikor a vevő egyedi címén kerül a csomag felvételre, majd a partnernek kiszállítjuk.
- Lehet háromszereplős (**ThirdPersonDelivery**), ahol egy harmadik szereplőnél kerül a csomag felvételre majd a címzettnek kiszállítjuk. Az értékesítési osztály engedélye szükséges ennek a csomagfeladási típusnak a használatához.

Az egyedi cím Országkód, Irányítószám, Város, Utca, Házzám adatokból épül fel.

A programkódban egyszerű paraméterezéssel lehet eldönteni, hogy melyik típus szerint szeretnénk rögzíteni a csomagjainkat. Ezen dokumentumban kifejtésre kerül, hogy melyik típushoz milyen kötelező beállítások szükségesek.

3.3.1. Nemzetközi házhozszállítás - FanCourier

Jelenleg Romániába van lehetőség a FanCourier futárszolgálattal házhoz szállítással csomagot feladni. Ilyenkor nem elég a **CountryCode**-ban a „RO” országkódot megadni, hanem a **SupplimentJSONData**-t is megfelelően ki kell tölteni a szállítóra specifikus adatokkal. Részletek a forráskódban találhatóak.

Ilyenkor a regisztrációs válaszban a **NewBarcode** mező mindig ki van töltve a FanCourier-től visszakapott csomagonkénti vonalkóddal (pl. 2046700120026001F0500). A **ShipmentID**-ban pedig az AWB szám (pl. 2046700120026) található, amely a fuvarlevélszámnak felel meg, tracking-ben is ez jelenik meg. A FanCourier saját címkét generál, ennek részleteit a címkegenerálás fejezetben



részletezzük, a csomagon ezt a dokumentumot kell elhelyezni. A válasz ErrorMessageInternational mezőjében található a FanCourier-től visszakapott hibaüzenet.

3.4. Vonalkód csere

A SIKERESEN BEREKISZTRÁLT CSOMAGOKNÁL VISSZAKAPUNK EGY NEWBARCODE MEZŐT IS, MELYBEN AKKOR JELENIK MEG AZ ÁLTALUNK GENERÁLT BARCODE-TÓL ELTÉRŐ VONALKÓD, HA MÁR A RENDSZERBEN KORÁBBAN SIKERESEN RÖGZÍTETTÜNK CSOMAGOT AZ ADOTT BARODE-DAL. ILYENKOR A RENDSZER GENERÁL EGY ÚJ, EGYEDI BARCODE AZONOSÍTÓT. EZT AZ AZONOSÍTÓT IS EL KELL TÁROLNI, HISZEN A KÉSŐBBIEKBEN A CSOMAGCÍMKÉT ERRE AZ ÚJ VONALKÓDRA TUDJUK LEKÉRNI. A TRACKING-BEN IS ERRE A VONALKÓDRA TUDUNK KERESNI.

„NewBarCode eltér az OriginalBarCode-tól” az alábbi esetekben fordulhat elő:

- A hívó fél egy már sikeresen regisztrált vonalkódot újra beküld.
- A generált, beküldött vonalkód, már létezik a rendszerben. (Más partner is ugyan azt a vonalkódot küldi el. Ennek kivédése érdekében, minden partner kap egy egyedi BarCodePrefix-et, amit be kell építeni a kódba.)
- Internet kapcsolati anomália vagy kiesés miatt, nem érkezik meg a **Response** időben (**TimeOut**), de a csomag a rendszerben rögzítésre került. Így a sikeres **Response** érdekében újra beküldésre kerül ugyan az a vonalkód. Amennyiben már tudjuk, hogy egyszer beküldtünk egy vonalkódot, és **TimeOut** miatt kaptunk hiba üzenetet, érdemes lekérni a csomag címkét vonalkód alapján. Amennyiben nincs ilyen címke, úgy a rendszerbe sem került berögzítésre a vonalkód.
- Házhoz szállításos „többcsomagos” regisztrációnál (ParcelCount>1) belekerülnek a kapcsolt csomagok vonalkódjai is (Barcode, Barcode-2, Barcode-3...)
- Nemzetközi szállításnál (pl. FanCourier) a külföldi logisztikai szolgáltató más vonalkódot oszt ki. (pl.: 2347600120021001F1200)

3.5. Okmány visszaforgatás

3.5.1. Pick Pack Pont

Abban az esetben, amikor a csomag átadást követően szeretnénk az átvevő ügyfél által aláírt szerződést/dokumentumot/iratot visszakapni. Erre kínál lehetőséget az okmányos csomag regisztráció. Okmányos csomagregisztráció használatba vételéhez előzetes egyeztetés, jogosultság szükséges. Ellenkező esetben az ilyen jellegű csomagfeladás sikertelen lesz. Ellenkező esetben az ilyen jellegű csomagfeladás sikertelen lesz. Okmány visszaforgatáskor a feladott csomagba az okmány mellé mellékelni kell egy borítékot és rá kell ragasztani a visszakapott Return csomag címkéjét is! A boltban ebbe fogják beletenni a dokumentumot és juttatják vissza a feladóhoz.

Okmány visszaforgatásos csomagok esetén a Normal/Pos2Pos/Direct csomag rögzítésekor a ReturnOfDocument mezőt true-ra kell állítani. Ilyenkor az adat beküldésekor a rendszerben automatikusan generálódik egy Return típusú csomag, amelynek a vonalkódja az eredeti csomag vonalkód, plusz egy „R” betű lesz. Ezt vissza is adjuk a válaszban. Behívás alapú címke generálás



esetén ezt a vonalkódot is be kell küldenia kérésben, majd a fentiek szerint a címkét el kell helyezni a csomagban.

Házhoz szállításos csomagok esetén nincs a fentiek szerinti megkötés.

3.5.2. Házhoz szállítás

Okmány visszaforgatás szolgáltatás igénybe vételéhez egy Parcel-t (fuvarlevelet) szükséges rögzíteni. Fontos különbség, hogy többcsomagos megrendelés esetén is van lehetőség okmány visszaforgatásra.

Amennyiben a ServiceType = HomeDeliver, és a ReturnOfDocument=true értékkel kerül rögzítésre, úgy a csomaghoz visszaküldendő okmányt csatolhatunk.

3.6. Típusok, mezők leírása

3.6.1. Kiemelt fontosságú mezők

Ezekre fontos odafigyelni, mert egy-egy téves beállítás, vagy szükséges beállítás kihagyása (amit a rendszer akár default értéként kezelhet) a csomag életútjában okozhat nem szándékolt változásokat. **A kérdéses mezőknél kérjük, egyeztessen az érintett értékesítővel, mivel az egyes opciók szerződéstől függőek, és eltérő díjazásúak lehetnek.**

3.6.1.1. ServiceType

Ez határozza meg az igénybe vett szolgáltatást, melyeknek eltérő szállítási módja(i) és díjazása lehet(nek). Pick Pack Pont szállításon belül 4 kategóriát különböztetünk meg: Normal, Pos2Pos, Direct, Return. A HomeDeliver-rel a házhoz szállítást lehet kiválasztani. Részletes leírás [itt](#).

3.6.1.2. TransitTime

Ezt a mezőt a PPP-s csomagoknál nem kell kitölteni, csak házhoz szállításnál, viszont ez az érték meghatározza a csomag szállítási idejét. Lehetséges értékek [itt](#).

3.6.1.3. ParcelCount

Házhoz szállításos csomagoknál itt lehet megadni, hogy hány csomagot kell egyben kézbesíteni. A méretükből, súlyukból adódóan előfordulhat, hogy fizikailag több csomagot szállítunk ki, de együtt kell kézbesíteni őket ugyanannak a címzettnek. Ilyenkor a háttérben megadott számú csomag jön létre a következő vonalkód formátummal: Barcode, Barcode-2, Barcode-3. Ezek megjelennek a válasz NewBarcode mezőjében az eredeti vonalkódhoz és külön csomagcímke generálódik hozzájuk.

3.6.1.4. ReturnOfDocument

Pick Pack Pontra feladott, okmány visszaforgatásos csomag rögzítés esetén ebben a mezőben **true** értéket kell megadni. A mező jelzi, hogy a csomagkiadást megelőzően, az ügyfélnek alá kell írnia a csomaghoz tartozó dokumentumot.

3.6.2. Függvények

RegisterParcelContainer

Visszatérési objektum típusa: `return RegisterParcelContainerResponse`
Paraméterek: `RegisterParcelContainerRequest request`
Kötelező elem: **Igen**

Típus: Függvény

Leírás:

Általános csomagregisztrálás, mely **RegisterParcelContainerRequest**-et vár értékként és visszaad **RegisterParcelContainerResponse** típusú adatot. Ebben a válasz üzenetben visszakapjuk a beregisztrált csomagok Vonalkódját, esetenként az új vonalkódot, valamint a rögzítés sikerességéről tájékoztató jellegű **ErrorCode**-ot.

3.6.3. Típusok

3.6.3.1. Request

RegisterParcelContainerRequest

Típus: Osztály

Leírás:

Partneri csomagregisztrációhoz szükséges Request-et megvalósító osztály. Partner azonosítóra, csomagok listájára és egy kivezetésre kerülő, de jelenleg kötelező Beszállítás módja mezőre van szüksége.

A megadott típusoknál hosszabb értékek automatikusan levágásra kerülnek!

Mezők:

Típus	Mezőneve	Megjegyzés	Kötelező
string (10)	PartnerCode	Partner (feladó) azonosító	Igen
string	PartnerAddress	Megszűnt mező. Beszállító címe.	Nem
List<Parcel>	ParcelContainer	Csomag adatok gyűjteménye. A regisztrálni kívánt csomagok listája. Az egyes Parcel elemek beállításától függ, hogy milyen csomagot regisztrálunk	Igen
DateTime	ArriveDate	Beszállítás várható ideje.	Nem

		Felületen nem jelenik meg.	
Supplier	Supplier	Megszűnő mező. A szállítványozók típusának megkülönböztetésére használt ENUM. Partneri Törzsadat szinten tárolt PPP- és HD beszállítás módja alapján állítódik. Jelenleg még kötelező Partner-ként kitölteni	Igen
string(10)	Token	Partnerek csomagfeladását hitelesítő mező. 10 karakter hosszú, kisbetűket és számokat tartalmaz!	IGEN
bool	OnlyLabelPrinting	Csak címke nyomtatás céljából történő adatbeküldés	Nem
bool	FinalizeLabelPrinting	Címke nyomtatás véglegesítése, csomag regisztráció beküldése	Nem
string(max)	SupplimentJSONData	Egyéb adatok	Nem

Parcel

Típus: Osztály

Leírás:

Csomag regisztrációnál, az egy csomagot megvalósító osztály. Felparaméterezéstől függően lehet beállítani, hogy milyen csomag típust szeretnénk rögzíteni.

Egy Parcelen belül maximum 100 csomagot lehet rögzíteni abban az esetben, ha Házhoz szállítási csomagot rögzítünk. Ezt a darabszámot a **ParcelCount** mezőben tudjuk megadni. Pick Pack Pontos csomagok esetében ennek a mezőnek a kitöltését a rendszer figyelmen kívül hagyja.

Mezők:

Típus	Mezőneve	Megjegyzés	Kötelező
dateTime	AccountingDeadlineDate	Számlafizetés határideje	Nem
string (28)	AutorizationCode	Megrendelés kódja, ami a partner saját csomag azonosítója, vagy a termék számlájának a száma.	IGEN
string (25)	BarCode	Csomag vonalkódja	IGEN
string (64)	ContactName	Kapcsolattartó neve	NEM
string (256)	CustomerAddress	Ügyfél címe (utca)	IGEN
string (25)	CustomerCity	Ügyfél címe (város)	IGEN
string (10)	CustomerCode	Címzett kódja, ha van	Nem
string (2)	CustomerCountryCode	Ügyfél ország kódja	NEM
string (64)	CustomerEmail	Ügyfél E-mail címe	IGEN
string (256)	CustomerName	Ügyfél Neve	IGEN
string (64)	CustomerPhone	Ügyfél Telefonszáma SMS tájékoztatáshoz	IGEN



		+36201234567 formátumban	
string (8)	CustomerPostalCode	Ügyfél Irányítószáma	IGEN
string (40)	CustomerRegion	Címzett régiója/megyéje	NEM
string (40)	CustomerStreetNumber	Címzett házszám plusz további részletek (emelet, ajtó) (kiegészíti a CustomerAddress-t)	IGEN
CustomerType	CustomerType	Ügyfél típusa (mindig B2C).	NEM
decimal(9,2)	DeliveryPrice	Megszűnt mező. Korábban a fuvardíjat jelölte, jelenleg a mező értéke a rendszerben számítható ki.	NEM
string(3)	DeliveryPriceCurrency	Fuvardíj valutája	NEM
ParcelDeliveryType	DeliveryType	Kézbesítési mód megválasztása.	HD esetén
string (100)	Description	Megjegyzés a csomaghoz és a futárnak	NEM
string (10)	DestinationLocationId	Cél PPP azonosítója, ahova a csomag kiszállítását kéri. Amennyiben a csomag célja a Lapker központi raktára, úgy a 2000-es kódot használja. ShopCode-ot vár	IGEN csak PPP esetén
string(28)	DirectPackageBarCode	Megszűnt mező. Direkt helyszínre szállítandó fizikai csomag vonalkódja.	NEM
string (64)	InvoiceNumber	Megszűnt mező. A csomagban lévő számla száma, átvette a helyét az AuthorizationCode	NEM
boolean	IsPartnerInvoiced	Partner felé kell-e a számlát kiállítani	NEM
decimal(9,2)	PackagePrice	Csomag értéke (Kártérítés esetében a terhelés alapja) Maximális összeg: 13m Ft. Csak egész számokat lehet megadni! A tizedesjegyeket a rendszer matematikai szabályok szerint kerekíti!	IGEN
string(3)	PackagePriceCurrency	Csomag értékének valutája, default HUF	IGEN
decimal(19,5)	PackageSizeX	Csomag méret adat	HD esetén
decimal(19,5)	PackageSizeY	Csomag méret adat	HD



			esetén
decimal(19,5)	PackageSizeZ	Csomag méret adat	HD esetén
PackageType	PackageType	Csomag méretszerinti típusa	PPP esetén
decimal(19,5)	PackageVolume	Csomag térfogata m ³	HD esetén
decimal(19,5)	PackageWeight	Csomag súlya kg-ban, maximum 20kg	HD esetén
string (64)	ParcelContentDescriptio n1	A csomag tartalmi leírása, Tartalmi megjegyzés	NEM
string (64)	ParcelContentDescriptio n2	Termék név	NEM
int	ParcelCount	Futár esetben együtt kézbesítendő Csomagok darabszáma Ebben a mezőben lehet megadni, az egy Parcel objektumon belüli csomagok darabszámát. PL: ParcelCount = 100 azt jelenti, hogy 100db csomagot regisztrálunk.	HD esetén
ParcelCreationStatus	ParcelCreationStatus	A csomag létrehozás folyamatának aktuális státusza. Default Create-el kell beküldeni.	IGEN
string (64)	PickupCustomerName	Feladó neve	NEM
string (10)	PickupLocationId	Csomagfeladás helye ShopCode-ot vár.	Pos2Pos esetén
string (64)	PickupNotificationEmail Address	Feladó email címe	NEM
string (32)	PickupNotificationPhone	Feladó telefonszáma	NEM
decimal(9,2)	PriceAtDelivery	Átvételkor fizetendő díj, amikor a vásárló átveszi. Maximális összeg: 700000 Ft. Csak egész számokat lehet megadni! A tizedesjegyeket a rendszer matematikai szabályok szerint kerekíti!	IGEN
string (3)	PriceAtDeliveryCurrency	Átvételkor fizetendő díj valutája, amikor a vásárló átveszi, default HUF	IGEN
string (28)	ReferenceBarCode	Referált vonalkód, elszámolásban megjelenik, Okmányvisszaforgatás esetén a rendszerben lesz hivatkozás a Return csomagra.	NEM



dateTime	RegisterDate	Feladás dátuma	NEM
boolean	ReturnOfDocument	Okmány visszaforgatás igény. PPP-s csomagok esetén a ReferenceBarCode kitöltése kötelező.	NEM
ParcelServiceType	ServiceType	Kiszállítási mód. Csomagregisztrációnál megadott érték alapján lehet eldönteni, hogy milyen szolgáltatást kell a csomagra értelmezni.	IGEN
string(max)	SupplimentJSONData	Egyéb, speciális adatok beküldésére szolgáló mező.	NEM
boolean	Tracking	Megszűnt mező. Tracking követés kérelmének jelzése	NEM
int	TransitTime	Szállítási idő Lehetséges értékek: 0 - 9 óra előtti 1 - délelőtt 9-12 óra között 2 - 1 munkanapos 3 - 2 munkanapos 4 - 3 munkanapos 5 - 5 munkanapos	HD esetén
ThirdPersonDelivery	ThirdPersonDelivery	Háromszereplős csomagfeladásnál a felvét helyének adatai	NEM

Supplier

Típus: Enumerátor

Leírás:

Beszállítás módja, ami alapján be lehet azonosítani, hogy a csomagot ki hozza be, illetve futárnak kell-e kitervezni kiszállást. A rendszerben ennek az enumerátornak a használata a későbbiekben kivezetésre kerül partneri törzsadat beállítás szerint. Jelenleg kötelező a kitöltése Partner értékkel.

Mezők:

Név	Megjegyzés
Partner	Partner hozza be a csomagot
Lapker	Lapker hozza be a csomagot
Customer	Sprinter hozza be a csomagot

CustomerType

Típus: Enumerátor

Leírás:

Az ügyfelek típusának megkülönböztetésére használt enum, B2C-vel kell kitölteni



Mezők:

Név	Megjegyzés
B2B	Business-to-Business típusú ügyfél
B2C	Business-to-Consumer típusú ügyfél
C2C	Consumer-to-Consumer típusú ügyfél

ParcelDeliveryType

Típus: Enumerátor

Leírás:

Kézbesítés módjának megkülönböztetésére használt felsorolós típus. Csak HomeDeliver ParcelServiceType esetén használendő.

Mezők:

Név	Megjegyzés
ThirdPersonDelivery	Háromszereplős kézbesítés.
DeliveryAndReturns	Csomagkézbesítés és visszárus csomagfelvétel
OnlyDelivery	Csak csomagkézbesítés
OnlyReturns	Csak visszárus csomagfelvétel
None	Default érték, a PUDO automatikusan OnlyDelivery-nek veszi

PackageType

Típus: Enumerátor

Leírás:

PPP csomagok esetén értelmezett csomag méret besorolás.

Mezők:

Név	Méret	Megjegyzés
Small	X: 20cm, Y: 30cm, Z: 10cm, Volume: 0.006000m ³	Kicsi
Medium	X: 30cm, Y: 30cm, Z: 20cm, Volume: 0.018000m ³	Közepes
Large	X: 50cm, Y: 50cm, Z: 50cm, Volume: 0.125000m ³	Nagy
Special	X: 60cm, Y: 60cm, Z: 60cm, Volume: 0.216000m ³	Speciális
None		Nincs megadva méret

ParcelCreationStatus

Típus: Enumerátor

Leírás:

A beküldött csomag feldolgozási módjának megadására szolgáló enum.

Mezők:

Név	Megjegyzés
Create	Csomag létrehozása

ParcelServiceType

Típus: Enumerátor

Leírás:

A csomag kiszállítási módjának megkülönböztetésére használt felsorolós típus. A különféle típusokra külön elszámolások keletkeznek.

Mezők:

Név	Megjegyzés
Normal	Normál PPP-s csomag, Raktárból történő feladás
Direct	PPP-n feladott, és ugyan abban a PPP-n átvehető csomag
Pos2Pos	PPP-n feladott csomag, melyet egy másik PPP-n vesznek át
Return	Visszárus csomag, melyet PPP-n adnak fel
HomeDeliver	Házhoz szállításos csomag, futárral történik a kiszállítás

ThirdPersonDelivery

Típus: Osztály

Leírás:

Háromszereplős csomagfeladásnál a csomagfeladó adatait tartalmazza. A kötelező mezőket csak abban az esetben kötelező kitölteni, ha háromszereplős csomagról van szó.

Mezők:

Típus	Mező neve	Megjegyzés	Kötelező
string	Name	Feladó neve	IGEN
string	CountryCode	Feladó ország kódja	IGEN
string	PostalCode	Feladó postai irányítószáma	IGEN
string	City	Feladó település	IGEN
string	Street	Feladó utca neve	IGEN
string	StreetNumber	Feladó házszám	IGEN
string	Floor	Feladó emelet	NEM
string	RoomNumber	Feladó lakás száma	NEM
string	EmailAddress	Feladó email címe	NEM
string	PhoneNumber	Feladó telefonszáma	NEM

3.6.3.2. Response

RegisterParcelContainerResponse

Típus: Osztály

Leírás:

Partneri csomagregisztrációhoz szükséges Response-t megvalósító osztály, mely tájékoztatást ad a beregisztrált csomagok sikerességéről esetenként, ha megváltozott a csomag vonalkód, ezt is visszakapjuk válasz üzenetben.

Mezők:

Típus	Mezőneve	Megjegyzés
string	ErrorCode	Hibakód
List<ParcelResult>	ParcelResults	Beregisztrált csomagokról visszajelzés, információ

ParcelResult

Típus: Osztály

Leírás:

Beregisztrált csomag, regisztrációjának sikerességéről válasz üzenet. Tartalmazza az általunk beküldött vonalkódot, esetenként visszakapjuk az új vonalkódot, illetve szállítmány azonosítót.

Mezők:

Típus	Mezőneve	Megjegyzés
string	ErrorCode	Hibakód
string	OriginalBarCode	Eredeti vonalkód, amivel be lett rögzítve a csomag
string	NewBarCode	Új vonalkód
string	OriginalDirectPackageBarCode	Megszűnt
string	NewDirectPackageBarCode	A fizikai csomag új vonalkódja (ha újat kellett generálni). Megszűnt.
string	ShipmentID	A csomaghoz tartozó Shipment azonosítója, amit a háttérrendszer generál. ServiceType-onként egy azonosító alá kerülnek a csomagok PPP esetén. Futár szállítás esetén minden Parcel-hez külön generálódik.
string	DirectionNumber	Kézbesítési címhez tartozó irányszám. Futáros irányítószám-irány, vagy PPP központi raktárból a cél felé induló járat.
string	DirectionNumberSender	Feladási címhez tartozó irányszám



		(Csak futár, visszáru esetén)
string	ErrorMessageInternational	Nemzetközi áthívásból jövő válaszüzenet
boolean	IsDelayedDelivery	Késleltetett kiszállítás. Megszűnt.

3.6.3.3. Hibakódok

ErrorCode

Típus: string

Leírás:

Lehetséges hibakódok és magyarázatuk.

Mezők:

Szállítmány szint

Név	Megjegyzés
PSR_OK	Beszállítói csomag, vagy szállítmány regisztrációja sikeres volt
PSR_FAILED	Beszállítói csomag, vagy szállítmány regisztráció nem sikerült ismeretlen hiba miatt.
PSR_PARTNERCODE_MISSING	Beszállítói azonosító nincs megadva
PSR_INVALID_PARTNER	Beszállítói azonosító nem található, vagy hibásan van beállítva
PSR_PARTNER_MISSING_COURIER_PREMISE	A megadott partnerhez nincsen futáros telephely beállítva, keresse az ügyfélszolgálatot!
PSR_PARTNER_NOT_FOUND	Nem található a megadott partner kód
PSR_EMPTY_SHIPMENT	Üres csomaglista!
PSR_SHIPMENT_NOTENUMBER_ALREADY_USED	Szállítmányazonosító ütközés
PSR_SHIPMENT_NOTENUMBER_MISSING	Szállítmányazonosító nincs megadva
PSR_SHIPMENT_NOTENUMBER_NOT_FOUND	Szállítmányazonosító nem létezik

Csomag szint

Név	Megjegyzés
PSR_OK	Beszállítói csomag, vagy szállítmány regisztrációja sikeres volt
PSR_FAILED	Beszállítói csomag, vagy szállítmány regisztráció nem sikerült ismeretlen hiba miatt.
PSR_INVALID_PACKAGE	A módosítandó csomag nincs meg!
PSR_INVALID_SHOPID	Az árushely kód hibás!
PSR_INVALID_PARTNER_SHOP_COUNTRY	A megadott partner nem adhat fel csomagot a kiválasztott boltba, mert nem szerepel a célországai között
PSR_PACKAGE_BARCODE_MISSING	Csomag vonalkód nincs megadva
PSR_PACKAGE_BARCODE_INVALID	Csomag vonalkód érvénytelen
PSR_SHIPMENTITEM_CANNOTMODIFY_ALREADY_PICKEDUP	Shipment nem módosítható, már beérkezett
PSR_PACKAGE_INVALID_CURRENCY	Csomag pénzneme nem megfelelő.
PSR_INVALID_TRANSIT_TIME	Szállítási idő érvénytelen
PSR_INVALID_PAYMENT_METHOD	Fizetési mód érvénytelen



PSR_INVALID_SUPPLIMENT_DATA	Érvénytelenül kitöltött suppliment adat
PSR_INVALID_SUPPLIMENT_DATA_DELIVERY_PRICE	Érvénytelenül kitöltött suppliment adat, delivery price
PSR_SHIPMENT_CUSTOMENAME_MISSING	Vevő neve nincs megadva
PSR_SHIPMENT_CUSTOMERPHONE_MISSING	Vevő telefonszáma nincs megadva
PSR_SHIPMENT_CUSTOMERPHONE_INVALID	Vevő telefonszáma helytelen. Nem helyes az előhívószám vagy a formátum, csak +36 és +40es számokat fogadunk el Regex szabály: ^((0036 [+36 36 06][0-9]{8,9}) (0040 [+40 40 07]7[0-9]{8}))\$
PSR_SHIPMENT_CUSTOMERADDRESS_MISSING	Vevő címe nincs megadva
PSR_INVALID_POSTALCODE	Megadott irányítószámhoz nem tartozik irány.
PSR_FAILED_POSTALCODE_MISMATCH	Hiba, amennyiben az irányítószám / város nem létezik, vagy nem egymáshoz tartoznak
PSR_FAILED_THIRD_PERSON_DENIED	Hiba, partner számára nem engedélyezett a háromszereplős csomagregisztráció.
PSR_FAILED_THIRD_PERSON_MISSING_DATA	Hiba, háromszereplős feladásnál minden adat megadása kötelező, kivéve emelet, ajtó.
PSR_FAILED_THIRD_PERSON_INVALID_POSTALCODE	Hiba, háromszereplős feladásnál a felvétel címében nem megfelelő irányítószám
PSR_FAILED_THIRD_PERSON_POSTALCODE_MISMATCH	Hiba, háromszereplős feladásnál a felvétel címében nem megfelelő irányítószám és helység páros
PSR_INVALID_DELIVERY_PARTNER	Nincs megadva vagy nem értelmezhető szállító partner.
PSR_EXTERNAL_SYSTEM_NOT_AVAILABLE	Csomag regisztrációhoz szükséges külső rendszer nem elérhető, később próbálja újra.
PSR_EXTERNAL_SYSTEM_UNKNOW_ERROR	Külső rendszer hívása során ismeretlen hiba történt, tovább részletek az InternationalErrorMessage mezőben.
PSR_EXTERNAL_SYSTEM_NOT_EXCEPTED_RESULT	Külső rendszer hívás során nem várt eredményt kaptunk, vegye fel a kapcsolatot az ügyfélszolgálatunkkal.
PSR_SHIPMENT_CUSTOMERCOUNTRY_MISSING	Vevő ország nincs megadva
PSR_SHIPMENT_CUSTOMERPOSTALCODE_OR_CUSTOMERR E G I O N _ R E Q U I R E D	Vevő postalcode vagy megye kötelező
PSR_INVALID_SERVICETYPE_DIRECT_NOT_POS2POS	Pos2Pos ServiceType-al adott fel olyan csomagot, melynek a cél és a feladási helye azonos. Az ilyen csomagot Direct-el kell feladni!
PSR_INVALID_PARTNER_TOKEN	Nem megfelelő a TOKEN. Nincs kitöltve, vagy hibásan lett megadva!
PSR_SHOP_NOT_ENABLED_RETURN_OF_DOCUMENT	Okmány visszaforgatás opció nincs engedélyezve a boltra. A bolt nem adhat ki ilyen csomagot.
PSR_PARTNER_NOT_ENABLED_RETURN_OF_DOCUMENT	Okmány visszaforgatás opció nincs engedélyezve a partnernek. A partner nem adhat fel ilyen csomagot.
PSR_RETURN_OF_DOCUMENT_NEEDED_REFERENCE_BARCODE	Okmány visszaforgatásos csomagnak nem lett megadva a referált vonalkód.



PSR_RETURN_OF_DOCUMENT_MISSING_ORIGINAL_PACKAGE	Okmány visszaforgatásos csomagot nem lehet rögzíteni, mert a referált csomag vonalkód nem rögzült/nem létezik a rendszerben.
PSR_TOO_MANY_PARCEL_IN_REQUEST	Túl sok csomagot próbált regisztrálni, maximum 1000-et lehet.
PSR_PACKAGE_AUTHCODE_MISSING	Csomag autorizációs kód nincs megadva.
PSR_PACKAGE_USED_BARCODE	Csomag vonalkód már használva volt. Új vonalkód szükséges.
PSR_INVALID_PRICE_CANNOT_BE_NEGATIVE	Az ár nem lehet negatív érték.
PSR_PACKAGE_INVALID_PRICE	A csomag értéke vagy utánvételi összege túl magas. Maximális értékek: Áruérték: 13 000 000 Ft Utánvét: 700 000 Ft
PSR_OK_PACKAGE_LABEL_PRINTING	Sikeres adatküldés címke nyomtatáshoz (OnlyLabelPrinting paraméterrel)! A csomag adat ilyenkor nem kerül feldolgozásra, csak a címke nyomtatáshoz rögzül adat.
PSR_PACKAGE_LABEL_PRINTING_NEEDED	Címke nyomtatás lezárást(FinalizeLabelPrinting) próbáltak kezdeményezni olyan csomagra, amire előzőleg nem küldtek címke nyomtatásra adatot (OnlyLabelPrinting)

4. Címke nyomtatás

4.1. Általánosan a címkegenerálásról

Ez a Service hívás csak azokra a csomag vonalkódokra ad vissza dokumentumot, melyeknél **PSR_OK** választ kaptunk a csomagrögzítését követően, vagyis ha a csomagfeladás rendben lezajlott. Ellenkező esetben a dokumentum előállítás hibaüzenetet ad vissza számunkra. Például nem létező, illetve nem rögzített csomag vonalkód esetén **WAR_PARCEL_NOT_EXISTS**, vagy **WAR_INVALID_BARCODE** hibaüzenetet kapunk. Amennyiben helyes csomag vonalkódot adtunk meg, úgy a válaszul kapott objektumban a RES_OK, illetve a DocumentName-ben a dokumentum neve szerepel.

A dokumentum neve a dokumentum beállításokból generálódik.

4.2. Címke nyomtatás csomag regisztráció nélkül

Alapesetben egy csomaghoz címkét nyomtatni (GetDocument service) csak akkor lehet, ha előzőleg a csomagot véglegesen regisztrálták (RegisterParcelContainer), ilyenkor a csomag bekerül a logisztikai rendszerbe, már nem módosíthatóak az adatai.

Egy másik opció, ha csak címke nyomtatásra van szükség (mivel a csomagok még csak összekészítés alatt vannak, és csak estére áll össze a végleges szállítmány), hogy a csomag adatokat megjelöljük a RegisterParcelContainer híváskor az **OnlyLabelPrinting=true** beállítással. Ha az adatküldés sikeres, akkor PSR_OK_PACKAGE_LABEL_PRINTING választ adunk vissza. Ilyenkor az adatokat rögzítjük, de még nem kerül be a logisztikai rendszerbe, viszont a GetDocument hívással már lehet címkét nyomtatni hozzá. A véglegesítésig ugyanezzel a paraméterrel a csomag adatok módosíthatóak.

Amikor befejezik a csomag előkészítését, akkor a RegisterParcelContainer-t újra meg kell hívni **FinalizeLabelPrinting=true** paraméterrel, és a Parcel listában azokkal a vonalkódokkal, amelyeket véglegesen be fognak küldeni. Ezzel az adatok bekerülnek a logisztikai rendszerbe, és már nem módosíthatóak.

A folyamat lépései:

1. Címke lekérés:
 - a. *RegisterParcelContainer* hívás **OnlyLabelPrinting=true** paraméterrel
 - b. Utána *GetDocument* hívás a címke lekéréséhez
2. Adatküldés véglegesítés:
 - a. *RegisterParcelContainer* hívás **FinalizeLabelPrinting=true** paraméterrel

4.3. Milyen dokumentumokat kaphat vissza

4.3.1. Címke/fuvarlevél

Ez a dokumentum típus a csomagok vonalkódja szerint generált címkéket adja vissza. Az egy oldalra nyomtatható címkék darabszámát a Size attribútum fogja állítani. Amennyiben nincs Size megadva, úgy az alapértelmezett méret a 2x2.

DocumentName: **DT_PackageLabel**

4.3.1.1. 2x2-es méret – A4-es lap 4 részre bontva

Ez a dokumentum a **Size = DS_2x2** beállítás után érhető el, illetve alapértelmezettként ez a beállítás érvényes. Ez a dokumentum egy A4-es lapot 4 részre oszt fel és ide generálja a csomag címkéket.

DocumentName: **DT_PackageLabel_DS_2x2**



Jelen képen a P_0 és P_1 pozíciókban elhelyezett csomagcímkek láthatóak, melyeket Házhoz szállítási csomagokra generáltunk le.

A HÁZHOZ SZÁLLÍTÁSOS ÉS PICKPACKPONT-OS CÍMKÉ GENERÁLÁSÁT A CSOMAG VONALKÓD ALAPJÁN DÖNTI EL A SERVICE, EZ NEM ÁLLÍTHATÓ PARAMÉTER.

A generált csomagcímkeken a következő adatoknak kell szerepelniük:

- Feladó adatai (Név, Irányítószám, Város, Utca, Telefon szám, Partner kód)
- Megjegyzés futárnak (ha nem rögzítettünk, akkor üres)
- Csomagadatok (vonalkód, darabszám, súly, áruérték)
- Kiszállítási idő
- Megrendelés kód, számlaszám
- Címzett adatai (Név, Irsz, Város, Utca, Telefon szám)
- Feladáskor és kézbesítéskor beszédendő összeg
- Utánvétel összeg
- 2D-s Datamatrix kód
- Sprinter és PPP logó
- Irány [HD vagy FX] járat szám



4.3.1.2. A6-os méret (Hőpapíros- és szalagnyomtatóhoz)

Ez a dokumentum a **Size = DS_A6** beállítás után érhető el. Ez a dokumentum tartalmilag és formailag teljesen megegyezik a 2x2-es címkével, viszont A6-os papírmérettel jön létre, így oldalanként 1 címke generálódik. Ez a formátum alkalmas Hőpapíros és szalagnyomtatóval történő nyomtatáshoz.

A helyes nyomtatáshoz az alábbi beállításokat kell elvégezni a nyomtatási beállításokban:

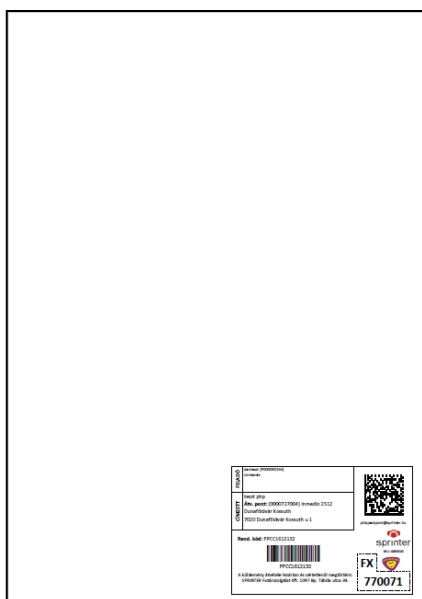
A6-os lapmérettel, Fekvő elrendezésben kell nyomtatni, előtte ellenőrizni kell a nyomtatási képet!

DocumentName: DT PackageLabel DS A6

4.3.1.3. 2x4-es méret – A4-es lap 8 részre bontva

Ez a dokumentum a **Size = DS_2x4** beállítás után érhető el. Ez a dokumentum egy A4-es lapot 8 részre oszt fel, és ide generálja a csomagcímkeket. Ez az opció csak PickPackPont-ra szállított csomagoknál érhető el.

DocumentName: **DT PackageLabel DS 2x4**



Jelen képen P 7 pozícióban elhelyezett csomagcímke látható.

A generált csomagcímkéken ugyanazoknak az adatoknak kell szerepelnie, mint a 2x2 címke esetében.

Elrendezésben és címke méretben tér el, illetve a feladó adatainál a feladó neve, azonosítója látható.

FELADÓ	eljuttatott: (P0000505244) címkeidév	 pickuppoint@sprinter.hu
	teszt.php Átv. pont: (0000727004) Inmedio 2512 Dunaföldvár Kossuth 7020 Dunaföldvár Kossuth u 1	
CÍMZETT		
Rend. kód: PPMC1612132  PPMC1612132 A küldemény átvétele lezártn és sértetlenül megőrtött. SPRINTER Futárszolgálat Kft. 1097 Bp. Táblás utca 39.		
		 sprinter 06-3-8036300 <div> <div>FX</div>  </div> <div>770071</div>

4.3.1.4. Nemzetközi FANCOURIER csomag címke

		Banca: _____ IBAN: _____	Valoarea declarata 3690 RON Ramburs Restituire	1 din
AWB Air Way Bill 2347600120021 13.12.2016				
EXPEDITOR FAN COURIER - cont test PERSONA DE CONTACT danteszt Fabrica de Glucoza (sosea), Nr. 11C, Bl. FAN, Cod postal: 020331 Bucuresti - 0 km adresa Bucuresti		PLIC 0	COLET 1	KG 1.00
DESTINATAR KAUFELD RESITA PERSONA DE CONTACT Balint Simai Bd. Muncii, Nr. 21 Resita - 0 km 108-Resita CONTINUT AS0901		ePOD PLATA LA expeditor PLATA TRANSPORT RAMBURS/RESTITUIE DOCUMENT DE PLATA CHITANTA		

Ez a dokumentum kizárólag román nemzetközi feladás során jön létre, a FanCourier futárszolgálat állítja elő. Ennek a dokumentum generálásához a Fan Courier-hez sikeresen bejegisztált csomag azonosítóra vagy AWB számra van szükségünk.

DocumentName: International_FANCOURIER_<<AWB_Number>>

4.3.2. Csomagátvételi elismervény

Ez a dokumentum típus a felvételkor átvételre kerülő csomagokat tartalmazza.

DocumentName: DT_DeliveryNote



Csomagátvételi elismervény - Fix pontos szállítás

Dátum 2016. december 14.



	Átadó		Átvevő
	P000050244	Átvevő neve:	Sprinter Futárszolgálat Kft.
Átadó neve:	danteszt	Átvevő címe:	Táblás utca 39
Átadó címe:	fő u 12	Település:	1097 Budapest
Csomag vonalkód	Címzett	Fuvar díj összeg	Utánvét összeg
PPCC1612132	teszt php		
Összesen			1 db Fix pontos szállítás

Átadó + bélyegző

Átvevő



Ez a dokumentum mind a Fixpontos szállítás, mind a Házhozszállítási csomagok esetében is így épül fel. Az egyetlen különbség, hogy Fix pontos dokumentum esetében nincs feltüntetve a Fuvardíj, illetve az Utánvét összeg.

4.3.2.1. Fixpontos szállítás – PPP

Ez a dokumentum tartalmazza a csomagátvételi elismervényt – Fixpontos szállítású csomag esetén. Ez a dokumentum A4-es méretben nyomtatódik ki.

A dokumentumon feltüntetésre kerül:

- **Átadó adatok** (Partneri adatok)
- **Átvevő adatok**
- **Csomag vonalkód** (ok)
- **Címzett(ek)**
- **Szállítás típusa(i)**
- **Kiállítás dátuma**

Továbbá a dokumentumon fel kell tüntetni kézírásos alapon az átvevő és átadó aláírását, továbbá az átadó bélyegzőjét is.

Ez a dokumentum nem tartalmaz Fuvardíj és Utánvét összeget.

DocumentName: **DT_DeliveryNote_PPP**

4.3.2.2. Házhoz szállítás – HD

Ez a dokumentum tartalmazza a csomagátvételi elismervényt – Házhozszállítási csomag esetén. Ez a dokumentum a dokumentum tartalmától függően egy A4-es lapra nyomtatódik ki.

A dokumentumon feltüntetésre kerül:

- **Partner adatok**
 - **Partner kód**
 - **Partner neve**
 - **Partner címe**
- **Átvevő adatok**
 - **Átvevő neve**
 - **Átvevő címe**
 - **Település**
- **Csomag vonalkód(ok)**
- **Címzett(ek)**
- **Fuvardíj összeg(ek)** (Csak Ft lehet!)
- **Utánvét összeg(ek)** (devizától függ)
- **Szállítás típusa(i)**



- Összesített adatok, **Fuvardíj összegről, Utánvét összegről, Szállítás típusának összegzéséről**
- Helyszínen beszedendő fuvardíj összesítő összeg
- Kiállítás dátuma

Továbbá a dokumentumon fel kell tüntetni kézirásos alapon az átvevő és átadó aláírását, továbbá az átadó bélyegzőjét is.

DocumentName: **DT_DeliveryNote_Courier**

4.3.3. Készpénzátvételi elismervény

Ez a dokumentum tartalmazza a készpénzes átvételi elismervényt mind a futár, mind a feladó részére. Ez a 2 dokumentum egy A4-es lapon jelenik meg, a lap közepénél jelzett elválasztó határolja el a dokumentumokat. Egy példány marad a futárnál, egy példány a feladónál.

A dokumentumon feltüntetésre kerül:

- Megrendelés kód
- Az átvett készpénz összege
- Város, Dátum a csomag átvételről.

Továbbá a dokumentumon fel kell tüntetni kézirásos alapon a Személy igazolvány számot, Átvevő nevét és Átadó nevét. Mind nyomtatott, mind aláírással formában.

DocumentName: **DT_ReceiptReport**



Megrendelés kód PPCC16120716

Sprinter példánya

SPRINTER CSOMAG ÉS LOGISZTIKA KÉSZPÉNZ ÁTVÉTELI ELISMERVÉNY

Alulírott Szem.ig.sz.: ezúton
elismerem, hogy a mai napon a danteszt –től átvettem 8788 Ft -ot a szállítási díjra*. A számla két hét múlva postai
úton érkezik a Feladó nevére és címére kiállítva.

Átvevő aláírása

Átadó aláírása

Átvevő aláírása nyomtatottan

Átadó aláírása nyomtatottan

Eger, 2016. december 14.

*A szállítási díj csak a házhoz szállítási szolgáltatási díjat foglalja magában. Ha fix pontra történt a megrendelés, akkor annak a házhoz szállítási díját nem fedezi.

Megrendelés kód PPCC16120716

Feladó példánya

SPRINTER CSOMAG ÉS LOGISZTIKA KÉSZPÉNZ ÁTVÉTELI ELISMERVÉNY

Alulírott Szem.ig.sz.: ezúton
elismerem, hogy a mai napon a danteszt –től átvettem 8788 Ft -ot a szállítási díjra*. A számla két hét múlva postai
úton érkezik a Feladó nevére és címére kiállítva.

Átvevő aláírása

Átadó aláírása

Átvevő aláírása nyomtatottan

Átadó aláírása nyomtatottan

Eger, 2016. december 14.

*A szállítási díj csak a házhoz szállítási szolgáltatási díjat foglalja magában. Ha fix pontra történt a megrendelés, akkor annak a házhoz szállítási díját nem fedezi.



4.4. Típusok, mezők leírása

4.4.1. Függvények

GetDocument

Visszatérési objektum típusa: `return MassDocumentResponse`
 Paraméterek: `MassDocumentRequest request`
 Kötelező elem: **Igen**

Típus: Függvény

Leírás:

Általános riportgenerálás, mely **MassDocumentRequest**-et vár értékként és visszaad **MassDocumentResponse** típusú adatot.

4.4.2. Típusok

4.4.2.1. Request

MassDocumentRequest

Típus: Osztály

Leírás:

Általános riportgenerálás, mely **MassDocumentRequest**-et vár értékként és visszaad **MassDocumentResponse** típusú adatot.

Mezők:

Típus	Mezőneve	Megjegyzés	Kötelező
List<BarcodeData>	Barcodes	Vonalkódok listája	Igen
List<DocumentSetting>	DocumentSettings	Dokumentum beállítások	Igen
string	SuplimentJSONData		Nem

BarcodeData

Típus: Osztály

Leírás:

Vonalkód, illetve a vonalkód típusát tartalmazó objektum.

Mezők:

Típus	Mezőneve	Megjegyzés	Kötelező
string	Barcode	Vonalkód	Igen
BarcodeType	Type	Vonalkód típusa	Igen
string	SuplimentJSONData		Nem

BarcodeType

Típus: Enumerátor

Leírás:

Vonalkód típust meghatározó enum

Mezők:

Név	Megjegyzés
BT_PackageBarcode	Csomag vonalkód
BT_ShipmentBarcode	Szállítmány vonalkód

DocumentSetting

Típus: Osztály

Leírás:

A legenerálandó dokumentumokat lehet vele felparaméterezni

Mezők:

Típus	Mezőneve	Megjegyzés	Kötelező
DocumentFormat	Format	DF_PDF vagy DF_ZPL	Igen
boolean	IsPositioned	true = Etikett címke false = normál A4-es lap	Nem
LabelDocumentPosition	Position	P_0 -tól P_7-ig	Nem
LabelDocumentSize	Size	2x2 vagy 2x4	Nem
string	SupplimentJSONData		Nem
DocumentType	Type	Címke, készpénz- vagy csomagátvételi elismervény, vagy mindegyik	Nem

DocumentFormat

Típus: Enumerátor

Leírás:

A válaszként adott dokumentum formátumát határozhatjuk meg vele. Ez lehet PDF fájl vagy Zebra nyomtatóknál használt ZPL kód.

Mezők:

Név	Megjegyzés
DF_PDF	PDF formátum megjelölése
DF_ZPL	ZPL kód megjelölése

LabelDocumentPosition

Típus: Enumerátor

Leírás:



Csomag címke nyomtatáskor, az első címke kezdő pozícióját lehet megadni.
Fentről lefele, balról jobbra haladva kell a pozíciók sorrendjét értelmezni.

Mezők:

Név	Megjegyzés	
	2x2	2x4
P_0	1	1
P_1	2	2
P_2	3	3
P_3	4	4
P_4	Ez a pozíció nem értelmezett	5
P_5		6
P_6		7
P_7		8

LabelDocumentSize

Típus: Enumerátor

Leírás:

A címkék méretének aszerinti meghatározása, hogy egy A4-es lapra hány darab fér rá.

2x2 = 4 db csomag címke

2x4 = 8 db csomag címke

Mezők:

Név	Megjegyzés
DS_2x2	A4-es lap, 2x2 részre történő felosztásban történik a címkék nyomtatása
DS_2x4	A4-es lap, 2x4 részre történő felosztásban történik a címkék nyomtatása

DocumentType

Típus: Enumerátor

Leírás:

Előállítandó dokumentum típusok

Mezők:

Név	Megjegyzés
DT_All	Minden típus megjelölése
DT_PackageLabel	Csomag címke
DT_DeliveryNote	Csomag átvételi elismervény
DT_ReceiptReport	Készpénz átvételi elismervény

4.4.2.2. Response

MassDocumentResponse

Típus: Osztály

Leírás:

Válaszüzenet a Szervertől. Ezt kapjuk vissza a **GetDocument()** metódustól.

Mezők:

Típus	Mezőneve	Megjegyzés
List<DocumentGenerationResult>	DocumentGenerationResults	Vonalkódonkénti státusz, sikerült-e a dokumentumgenerálás
List<DocumentData>	Documents	Vonalkódokhoz tartozó dokumentum
DocumentResults	Result	Válasz eredmény a teljes generálás sikerességéről
string	SuplimentJSONData	

DocumentResults

Típus: Enumerátor

Leírás:

Tájékoztatást ad, hogy a dokumentum előállítása sikeres volt, vagy valami hiba történt, például az adott csomag vonalkód nem létezik.

Mezők:

Név	Megjegyzés
RES_OK	Sikeresen megtörtént a dokumentumok előállítása
RES_DATA_MISSING	Hiányzó adatok miatt nem lehet előállítani dokumentumokat
RES_GENERATE_ERROR	Dokumentum előállítás közben nem várt hiba történt

DocumentGenerationResult

Típus: Osztály

Leírás:

Ez az objektum ad tájékoztatást arról, hogy az adott csomag vonalkódhoz sikerült-e dokumentumot generálni. Amennyiben nem sikerült, úgy erről a Result mező értéke tájékoztat.

Mezők:

Típus	Mezőneve	Megjegyzés
string	DocumentName	Az előállított dokumentum nevét



		tartalmazza, a DocumentSettings mező értékei alapján.
ResultCode	Result	Dokumentum létrehozásának sikerességéről tájékoztat
string	ResultMessage	
string	SuplimentJSONData	

ResultCode

Típus: Enumerátor

Leírás:

Tájékoztatót ad, hogy az egyes dokumentumok előállításuk sikeres volt-e vagy valami hiba történt, például az adott csomag vonalkód nem létezik, érvénytelen vonalkód került beküldésre.

Mezők:

Név	Megjegyzés
RES_OK	Sikeresen megtörtént a dokumentumok előállítása
ERR_NOT_IMPLEMENTED_DOCUMENT	Nem megvalósítható dokumentum
WAR_INVALID_BARCODE	Érvénytelen vonalkód
WAR_PARCEL_NOT_EXISTS	Nem létező csomag
WAR_PARCEL_NOT_PRESENTED_IN_ANY_REPORT	Riportban nem megjelenő csomag
ERR_DOCUMENT_GENERATION_FAILED	Hiba a generálás közben
WAR_SHIPMENT_NOT_EXISTS	Nem létező szállítmány

DocumentData

Típus: Enumerátor

Leírás:

Abban az esetben, ha DocumentGenerationResult.Result értéke **RES_OK**, akkor ez az objektum tartalmazza a legenerált dokumentumot a Document mezőben vagy a legenerált ZPL kódot a DocumentString mezőben.

Mezők:

Típus	Mezőneve	Megjegyzés
base64Binary	Document	Ez a byte tömb tartalmazza magát az előállított dokumentumot.
String	DocumentString	Ez a string tartalmazza a ZPL kódot, ha a DocumentFormatnál DF_ZPL-t adtunk meg.
string	DocumentName	Az előállított dokumentum nevét tartalmazza, a DocumentSettings mező értékei alapján
boolean	IsPositioned	Etikett címke, vagy

		sem
string	SupplimentJSONData	
DocumentType	Type	Az előállított dokumentum típusa

4.4.2.3. Hibakódok

A teljes generálásra vonatkozó lehetséges válaszok a MassDocumentResponse-on található DocumentResults leírásában találhatók.

Az egyes dokumentumok generálására adott válaszok pedig a DocumentGenerationResult-ban lévő ResultCode leírásában kerülnek felsorolásra.

5. Forráskód demó

A kód tartalmazza az összes csomagfeladási alaptípust mintaadatokkal, ezek helyére kell majd értelemszerűen behelyettesíteni a küldeni kívánt adatot.

Csomagregisztráció interfész teszt elérés:

<https://tsx.lapker.hu/PudoTest/PartnerPudoService?singlewsdl>

Dokumentumgenerálás interfész teszt elérés:

<https://tsx.lapker.hu/PudoTest/PudoDocumentService?singlewsdl>

5.1. C# kód

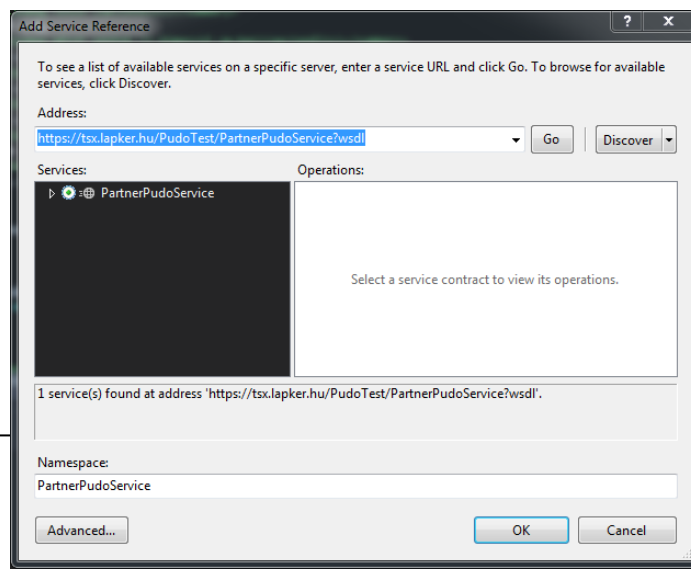
A példa kódot úgy állítottuk össze, hogy a korábban tárgyalt csomagregisztráció, illetve a - címkenyomtatás vonalkód alapján - kódrészeket egybe dolgoztuk.

Interface-t megvalósító .Net forrás: a mellékletben található az interface-t leíró dokumentum mellett. csharp/ParcelRegistration/ParcelRegistrationPUODOClient.sln

Az sln Visual Studio 2015 megnyitásával betölti a lentebb olvasható példa kódot.

5.1.1. Csomagregisztráció

Visual Studio projektünkön belül, adjuk hozzá a references –hez Add Service Reference segítségével a



PartnerPudoService Interface elérési útvonalát.

Miután sikeresen hozzáadtuk, a Solution Explorerben megjelenik egy új Service References mappa. Elkezdhetjük használni és implementálni a Service References által betöltött osztályt.

A CSOMAGRÖGZÍTÉS SORÁN VANNAK OLYAN KÖTELEZŐ ÉRTÉKEK, AMELYEK NEM ELHAGYHATÓAK, HIÁNYUK A SIKERES CSOMAGRÖGZÍTÉST A HELYES ADATOK MEGADÁSA ESETÉN IS MEGAKADÁLYOZHATJÁK. ILYEN FONTOS ADAT AZ EGYEDI VONALKÓD PREFIX MEGHATÁROZÁSA, EZÉRT A KÖVETKEZŐ KÓDRÉSZLETBEN ERRE IS MUTATUNK PÉLDÁT.

AZ EGYEDI AZONOSÍTÓ (PREFIX) PARTNERENKÉNT ELTÉR. AZ ELŐTAGOT A SPRINTER IT SUPPORT-TÓL KELL IGÉNYELNI!

Egy kialakított sémát követve készült el a demó forráskód, melyből a dokumentumba is beillesztésre kerültek részletek. Ennek a gondolatmenetnek köszönhetően épül fel a program végrehajtási logikája is. Azért, hogy egységesebb legyen a programkód magyarázata, a különböző eljárások közötti kommunikáció, nem szegmentáljuk.

Első lépésben építsük fel a regisztrálni kívánt csomagokat.

```
//berögztíteni kívánt csomagok listájának inicializálása  
List<Parcel> _parcelContainer = new List<Parcel>();
```

A korábban látott PPP, illetve HD felbontásokban folytatólagosan, külön-külön részletezzük a paraméterezést.

Ezekre a mezőkre azért kell odafigyelni, mert a különböző csomagrögztítéseket nem eltérő Objektumokkal valósítjuk meg. A különböző esetek miatt nem lehet megkötni 1-1 speciális, de fontos mező kitöltésének kötelezővé tételét.

Miután létrehoztuk a kívánt mennyiségű Parcel-t és odafigyeltünk a helyes adatok kitöltésére, össze kell rakni a Request-et és be kell küldeni a PUDO-nak.

Egy ParcelContainer MAXIMUM 1000 Parcel-t tartalmazhat! Ennél többet a rendszer nem fogad el.

```
RegisterParcelContainerRequest _registerParcelContainerRequest = new  
RegisterParcelContainerRequest()  
{  
    PartnerCode = "0000001000",           //Partnerazonosító  
    ParcelContainer = _parcelContainer,    //csomagadatok listája  
    Supplier = Supplier.Partner,          //Ez a sor kivezetésre fog kerülni, de  
                                           //jelenleg kötelező a kitöltése  
    Token = "testtoken1",                 //Csomagregisztrációt validáló Token  
    //ArriveDate = new DateTime(2017,1,1,18,0) //Beszállítás várható ideje, információs  
                                           //jellegű, opcionális  
};  
  
// kliens kapcsolat létrehozása és a korábban beállított csomagok elküldése/berögztítése  
PartnerPudoClient _partnerPudoClient = new PartnerPudoClient();  
// csomagrögztítés sikerességéről válasz üzenet  
RegisterParcelContainerResponse _registerParcelContainerResponse =  
    _partnerPudoClient.RegisterParcelContainer(_registerParcelContainerRequest);
```



A `RegisterParcelContainerResponse` válaszüzenetben megkapjuk a csomagok regisztrálásának sikerességét. Ezt első körben a `_registerParcelContainerResponse.ErrorCode` mezőből tudjuk meg. Illetve csomagonkénti lebontásban a `_registerParcelContainerResponse.ParcelResults.ErrorCode` ad tájékoztatást, hogy több csomag esetén melyik sikerült, melyik nem. A válasz üzenetben szereplő lehetséges `ErrorCode` üzeneteket feldolgozását később részletezzük.

5.1.1.1. Hibakezelés és adatfeldolgozás

Ez a rész azt dolgozza fel, hogy a válaszüzenetben kapott `ErrorCode` tájékoztató jellegű mező értékét hogyan értelmezzük.

Ha a mező értéke **PSR_OK**, akkor sikeres vagy csak részben sikeres volt a csomag / csomagok regisztrációja, de maga a szállítmány regisztrálása sikerült. Amennyiben nem **PSR_OK** ez az `ErrorCode`, akkor egyáltalán nem sikerült a csomagok regisztrációja és így tájékoztató jellegű is ez az üzenet. Példának okáért nem kerültek átküldésre a csomag adatok, így a válasz **ErrorCode**-ban **PSR_EMPTY_SHIPMENT** üzenetet kapunk.

A lehetséges hibaüzenetek korábban, a Hibakódok szekcióban részletezésre kerültek. Most megmutatjuk, hogy a sikeres csomagregisztráció esetén mi a teendő.

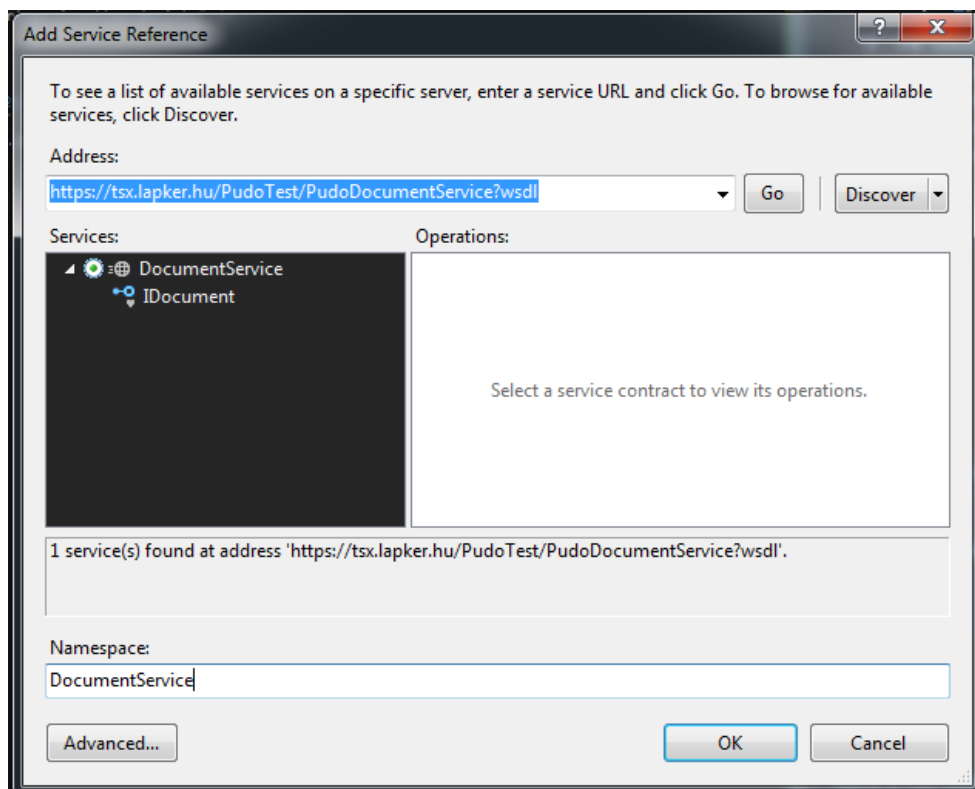
```
// Amennyiben a Response ErrorCode értéke "PSR_OK", úgy van értelme tovább
vizsgálni a Response-t
if(_registerParcelContainerResponse.ErrorCode == "PSR_OK")
{
    //Ebbe a listába tudjuk kigyűjteni a sikeresen berögzített csomagok
    vonalkódját, így később ezekre a vonalkódokra tudjuk lekérdezni a
    csomagcímkeket.
    List<BarcodeData> _barcodes = new List<BarcodeData>();
    // Haladjunk végig a ParceResults mezőn, ebben találjuk meg csomagonkénti
    lebontásban a csomagregisztrációhoz tartozó válasz adatokat.
    foreach(ParcelResult results in
        _registerParcelContainerResponse.ParcelResults)
    {
        Console.WriteLine($"OriginalBarcode : {results.OriginalBarCode}");
        Console.WriteLine($"NewBarcode : {results.NewBarCode}");
        Console.WriteLine($"ErrorroCode : {results.ErrorCode}");
        Console.WriteLine($"ShipmentID : {results.ShipmentID}");
        //Nemzetközi csomagregisztráció esetén
        if (!string.IsNullOrEmpty(result.ErrorMessageInternational))
        {
            Console.WriteLine($"ErrorMessageInternational
                                :{result.ErrorMessageInternational}");
        }
        Console.WriteLine();
        //Ha PSR_OK az adott csomag ErrorCode értéke, abban az esetben
        jegyezzük meg a vonalkódot.
        if (result.ErrorCode == "PSR_OK")
        {
            _barcodes.Add(new BarcodeData()
            {
                Barcode = result.NewBarCode,
                Type = BarcodeType.BT_PackageBarcode
            });
        }
    }
}
```

Így végig vittük a csomagregisztrációt és a Response-ban kapott válaszból tájékozódhatunk a csomagok beregisztálásának sikerességéről.

A példa kód összeállításánál figyelmet fordítottunk a csomagregisztráció, és vonalkód alapú dokumentum generátor kódrészek egybedolgozására.

5.1.2. Címke generálás

VisualStudio projektünkön belül, adjuk hozzá a references –hez Add Service Referenc segítségével a PUDODocumentService Interface elérési útvonalát.



Miután sikeresen hozzá adtuk a Solution Explorerben megjelenik egy új Service References mappa. Ezután hozzuk létre a vonalkódunkat tartalmazó listát.

```
//vonalkódokat tartalmazó lista
List<BarcodeData> _barcodes = new List<BarcodeData>();
_barcodes.Add(new BarcodeData()
{
    Barcode = "2347600120021001F1200", //kötelező megadni a vonalkódot
    Type = BarcodeType.BT_PackageBarcode //Kötelező beállítani a Type-ot
});

//Bármelyik két mező hiánya, exception-t okozhat
_barcodes.Add(new BarcodeData()
{
    Barcode = "PPCC1612131",
    Type = BarcodeType.BT_PackageBarcode
});
```

Barcode és Type kitöltése kötelező lépés, **nem elhagyható a Service hívás sikeressége miatt**. Adjuk meg a nyomtatandó dokumentumok beállításait.

```
List<DocumentSetting> _documentSettings = new List<DocumentSetting>();
_documentSettings.Add(new DocumentSetting()
{
    Format = DocumentFormat.DF_PDF,
    IsPositioned = false,
    Position = LabelDocumentPosition.P_0,
    Size = LabelDocumentSize.DS_2x2,
    Type = DocumentType.DT_All
});

_documentSettings.Add(new DocumentSetting()
{
    Format = DocumentFormat.DF_PDF,
    //IsPositioned = false,
    //Position = LabelDocumentPosition.P_0,
    //Size = LabelDocumentSize.DS_2x2,
    //Type = DocumentType.DT_All
});
```

MINIMUM 1 DOCUMENTSETTING() ELEMET HOZZÁ KELL ADNI A DOCUMENTSETTINGS LISTÁHOZ!

Vegyük észre a két dokumentum beállítás között lévő különbséget és egyezést.

Format = DocumentFormat.DF_PDF megadása kötelező elem, nélküle **exception**-t kapunk.

A többi mező értékének beállítása nem kötelező elem, mivel a Service automatikusan a **default** értékeket kezeli, ha nincs megadva.

Size	=	2x2, vagyis A4-es lapra 4 címke nyomtatható
Position	=	0 vagyis kezdő pozíció, bal felső saroktól kezdve
Type	=	minden dokumentum típus (csomag címke, készpénz és csomag átvételi elismervény)

```
//adjuk meg a request-nek a korábban beállított adatokat (vonalkód, dokumentum
beállítások)
MassDocumentRequest _massDocumentRequest = new MassDocumentRequest();
_massDocumentRequest.Barcodes = _barcodes;
_massDocumentRequest.DocumentSettings = _documentSettings;

//hozzuk létre a klienst
DocumentClient _documentClient = new DocumentClient();
//Majd a GetDocument fv-nek adjuk át a request-et
MassDocumentResponse _massDocumentResponse =
    _documentClient.GetDocument(_request);
```

Az így létrehozott példány **GetDocument()** metódusát meg lehet hívni, és visszakapjuk a választ, amiben benne lesz a vonalkódokhoz tartozó dokumentumok és **Result** kódjuk.

5.1.2.1. Hibakezelés és adat feldolgozás

`_response.Result == DocumentResults.RES_OK` esetén sikeres volt a service hívás, és így fel tudjuk dolgozni a `response.Documents` objektumban szereplő adatokat.

```
switch (_massDocumentResponse.Result)
{
    //Amennyiben a Result RES_OK, a dokumentumgenerálás sikeresen lefutott
    // és kaptunk vissza nyomtatható dokumentumot
    case DocumentResults.RES_OK:
    {
        //a response sikeres volt, fel lehet dolgozni a legenerált dokumentumokat.
        //Példa kód lentebb
        ...
        break;
    }
    case DocumentResults.RES_GENERATE_ERROR:
    {
        Console.WriteLine("Hiányzó adatok miatt nem lehet előállítani dokumentumokat");
        break;
    }
    case DocumentResults.RES_DATA_VALIDATION_ERROR:
    {
        Console.WriteLine("Nem sikerült a beküldött adatok validálása");
        break;
    }
    case DocumentResults.RES_DATA_MISSING:
    {
        Console.WriteLine("Dokumentum előállítás közben hiba történt");
        break;
    }
    default:
    {
        Console.WriteLine(_response.Result);
        break;
    }
}
```

RES_OK esetén az alább látható módon lehet feldolgozni a dokumentumokat.


```
//Válasz üzenetek a dokumentumokról
foreach (DocumentGenerationResult _documentGenerationResult in
_maddDocumentResponse.DocumentGenerationResults)
{
    Console.WriteLine(_documentGenerationResult.DocumentName);
    Console.WriteLine(_documentGenerationResult.Result);
    if (!String.IsNullOrEmpty(_documentGenerationResult.ResultMessage))
    {
        Console.WriteLine(_documentGenerationResult.ResultMessage);
    }
    Console.WriteLine();
}
int _index=0; //Ez a számláló segít abban, hogy az azonos nevű dokumentumok ne írják egymást felül
//FileStream lemezre írásakor
//Egyszerűsített dokumentum lemezre írása
foreach (DocumentData _document in _response.Documents)
{
    using (FileStream savefile = new
    FileStream($"C:/_{_document.DocumentName}_{_index++}.pdf",
    FileMode.OpenOrCreate))
    {
        savefile.Write(_document.Document, 0, _document.Document.Count());
        savefile.Flush();
    }
}
```

Ez a pár sor segítség a visszakapott adatok feldolgozásában.

Ezek alapján sikeresen elkészíthető a C# forrás kód, mely Valid csomag vonalkódok alapján visszakapja a kinyomtatható csomagcímkeket, készpénz- és csomagátvételi elismervényeket.

5.2. PHP kód

A példa kódot úgy állítottuk össze, hogy a korábban tárgyalt csomagregisztráció, illetve a - címkenyomtatás vonalkód alapján - kódrészeket egybe dolgoztuk.

5.2.1. Csomagregisztráció

Interface-t megvalósító php forrás: a mellékletben található az interface-t leíró dokumentum mellett. `php/ParcelRegistration/PartnerPudoService.php`

Teszt kód, mely használja az Interface-t:
`php/ParcelRegistration/PudoService.php`

A fenti, letölthető .zip fájlban található php osztály leírást töltsük le és fűzzük fel a PHP forráskódunkba.

Majd készítsük el a Service híváshoz szükséges fejléc adatokat, melyek egy biztonságos SSL kapcsolat titkosítását írják le és elengedhetetlen a kapcsolat sikerességéhez.

```
include("PartnerPudoService.php");  
//Ez az opció a https-en keresztüli TLS kapcsolat titkosítási módját állítja be. Ennek hiányában nem lehet  
behívni a serverre.  
$soap_options = array(  
    'features' => SOAP_SINGLE_ELEMENT_ARRAYS,  
    'stream_context' =>  
        stream_context_create(  
            array( 'ssl' => array(  
                'protocol_version' => 'tls1',  
                'ciphers' => 'DES-CBC3-SHA:RC4-SHA:RC4-MD5'  
            )  
        )  
    ),  
    'cache_wsdl' => WSDL_CACHE_NONE  
);  
$wsdl = "https://tsx.lapker.hu/PudoTest/PartnerPudoService?wsdl";
```

Ezután hozzuk létre a **request** kérésünket és töltjük ki a megfelelő adatokkal!

Első lépésben érdemes meghatározni a beregisztrálni kívánt csomagunk egyedi vonalkód azonosítóját.

Majd definiáljuk a rögzíteni kívánt csomagok listáját.

```
//Egyedi vonalkód Prefix, amit kaptunk  
$BarcodePrefix = 'ASD'.date("YmdHis"); // => ASD20170101010101  
//Csomagok gyűjtő listája  
$ParcelContainer = array();
```

Miután létrehoztuk a kívánt mennyiségű Parcel-t és odafigyeltünk a helyes adatok kitöltésére, össze kell rakni a Request-et és be kell küldeni a PUDO-nak.

Egy ParcelContainer MAXIMUM 1000 Parcel-t tartalmazhat! Ennél többet a rendszer nem fogad el.

```
// Hozzuk létre a Request  
$RegisterParcelContainerRequest = new RegisterParcelContainerRequest();  
$RegisterParcelContainerRequest->PartnerCode = "0000001000";  
$RegisterParcelContainerRequest->ParcelContainer = $ParcelContainer;  
$RegisterParcelContainerRequest->Token = "teszt0k3n"; //Csomagregisztrációt validáló Token  
//Itt adhatjuk meg a tervezett csomagfelvételének dátumát  
//$RegisterParcelContainerRequest->ArrivedDate = date("Y.m.d H:i:s"); //DEFAULT NULL  
  
// Hozzuk létre a Klient  
$PartnerPudoServiceClient = new PartnerPudoService($wsdl, $soap_options);  
  
// Küldjük el a Request-et és a Response választ ezután tudjuk feldolgozni  
$RegisterParcelContainerResponse =  
    $PartnerPudoServiceClient->RegisterParcelContainer($RegisterParcelContainerRequest);
```

A RegisterParcelContainerResponse válaszüzenetben megkapjuk a csomagok regisztrálásának sikerességét. Ezt a \$RegisterParcelContainerResponse->RegisterParcelContainerResult->ErrorCode mezőből tudjuk meg. Csomagonkénti lebontásban a \$RegisterParcelContainerResponse->RegisterParcelContainerResult->ParcelResults->ErrorCode ad tájékoztatást, hogy több csomag esetén melyik sikerült, melyik nem. A válaszüzenetben szereplő lehetséges ErrorCode üzeneteket később részletezzük.

5.2.1.1. Hibakezelés és adatfeldolgozás

Ebben a fejezetben részletezzük, hogy a válaszüzenetben kapott ErrorCode mező értékét hogyan értelmezzük.

Ha a mező értéke **PSR_OK**, akkor sikeres vagy részben sikeres volt a csomag, csomagok regisztrációja. A szállítmány regisztrálása sikerült. Amennyiben az ErrorCode nem PSR_OK, akkor egyáltalán nem sikerült a csomagok regisztrációja. Abban az esetben, ha nem kerültek átküldésre a csomag adatok, úgy a válasz **ErrorCode**-ban **PSR_EMPTY_SHIPMENT** üzenetet kapunk.

A lehetséges hibaüzenetek korábban a Hibakódok fejezetben részletezésre kerültek. Most megmutatjuk, hogy a sikeres csomagregisztráció esetén mi a teendő.

```
//Szállítmány szinten történő hibakódok feldolgozása
switch ($RegisterParcelContainerResponse->RegisterParcelContainerResult->ErrorCode) {
    case 'PSR_OK':{
        //Sikeres csomagregisztráció esetén ebben az ágba lehet feldolgozni a válasz üzenetben kapott
        //adatokat.
        //Érdemes a csomag címkékhez tartozó adatokat itt feldolgozni illetve ebben a részben
        //érdemesebb a csomag címke lekérést is elvégezni, amit szintén a későbbiekben be is mutatunk.
        break;
    }
    default:{
        echo $RegisterParcelContainerResponse->RegisterParcelContainerResult->ErrorCode;
        break;
    }
}
```

PSR_OK esetén az alább látható módon lehet feldolgozni a response szereplő csomag szintű adatokat.

```
//Sikeresen berögzített csomag vonalkódok kigyűjtéséhez szükséges tömb
$Barcodes = array();

//Nézzük végig csomagszinten a válaszüzenetben szereplő adatokat
$ParcelResults = $RegisterParcelContainerResponse->RegisterParcelContainerResult->ParcelResults;
foreach ($ParcelResults->ParcelResult as $Parcel) {
    echo ("Original Barcode : $Parcel->OriginalBarCode <br>");
    echo ("New Barcode : $Parcel->NewBarCode <br>");
    echo ("Error Code : $Parcel->ErrorCode <br>");
    echo ("ShipmentID : $Parcel->ShipmentID <br>");
    //Amennyiben nemzetközi csomagot regisztráltunk, az alábbi változóban lesz érték.
    if (!is_null($Parcel->ErrorMessageInternational) || $Parcel->ErrorMessageInternational!=""){
        echo ("ErrorMessageInternational : $Parcel->ErrorMessageInternational <br>");
    }
    //Ha csomaghoz tartozó ErrorCode PSR_OK, úgy sikeres volt a csomag regisztráció. Ennek a
    vonalkódját érdemes kigyűjteni a későbbi csomag címkelekérdezéséhez szükségünk lehet rá
    if($Parcel->ErrorCode=='PSR_OK'){
        $Barcodes[] = new BarcodeData($Parcel->NewBarCode);
    }
}
```

Így végig vittük a csomag regisztrációt és a **Response**-ban kapott válaszból tájékozódhatunk a csomagok beregisztrálásának sikerességéről.

5.2.2. Címke generálás

Interface-t megvalósító php forrás:

php/ParcelRegistration/**PudoDocumentService.php**

Teszt kód, mely használja az Interface-t:

php/ParcelRegistration/**PudoService.php**

A fenti, letölthető .zip fájlban található PHP osztály leírást töltsük le, és fűzzük fel a PHP forráskódunkba.

Majd készítsük el a Service híváshoz szükséges fejléc adatokat, melyek egy biztonságos SSL kapcsolat titkosítását írják le és elengedhetetlen a kapcsolat sikerességéhez.

```
include("PudoDocumentService.php");
//Ez az opció, a https-en keresztüli TLS kapcsolat titkosítási módját állítja be. Ennek hiányában nem lehet
behívni a serverre.
$soap_options = array(
    'features' => SOAP_SINGLE_ELEMENT_ARRAYS,
    'stream_context' =>
        stream_context_create(
            array( 'ssl' => array(
                'protocol_version' => 'tls1',
                'ciphers' => 'DES-CBC3-SHA:RC4-SHA:RC4-MD5'
            )
        )
    ),
    'cache_wsdl' => WSDL_CACHE_NONE
);
$wsdl = " https://tsx.lapker.hu/PudoTest/PudoDocumentService?wsdl ";
```

Ezután hozzuk létre a **request** kérésünket és töltjük ki megfelelő adatokkal.

Szükséges adatok:

- Vonalkódok listája

```
// vonalkód lista létrehozása, Példányosításkor kötelező vonalkódot megadni!  
$Barcodes[] = new BarcodeData("PPCC1612131"); //PPP  
$Barcodes[] = new BarcodeData("PPCC1612132"); //PPP
```

Egy tömb, aminek minden elemét a BarcodeData osztály példánya tölti ki. Rögtön a konstruktorból be lehet állítani a vonalkódot, ez egy elhagyhatatlan paraméter. Ennek hiányában nem fut tovább a php értelmező. Viszont itt a C# kóddal ellentétben, a Type mezőt is beállítja a php-s konstruktor, így ennek beállításával nem foglalkozunk. Kivéve, ha a PudoDocumentService.php fájlban sérült, vagy módosult a konstruktor. Ebben az esetben a konstruktorban be kell állítani a **\$this->Type = "BT_PackageBarcode"**; sort.

- Dokumentum beállítások listája

```
//documentSettings  
$documentSetting = new DocumentSetting();  
$documentSetting->IsPositioned = false; // Etikett? true | false  
$documentSetting->Position = LabelDocumentPosition::$P_0; // pozíció 0-7 2x4 esetén 0-3 2x2 esetén  
$documentSetting->Size = LabelDocumentSize::$DS_2x2; // címke méret, A4-es lap felosztása  
$documentSetting->Type = DocumentType::$DT_All; // generálandó dokumentum típusok  
// DT_All  
// DT_PackageLabel => csomag címke  
// DT_DeliveryNote => csomag átvételi elismervény  
// DT_ReceiptReport => kézpénz átvételi elismervény  
  
//az elkészült dokumentum beállításokat fűzzük hozzá a $DocumentSettings listához  
$DocumentSettings[] = $documentSetting;
```

A különböző mezők beállítása elhagyható, ha nincs megadva, vagy **null** értéket vesz fel, a service hívás a **default** értékkel dolgozik tovább. Így nem kapunk hibaüzenetet. **Csak arra kell figyelni, hogy a dokumentum beállításokat, még ha csak egyet is hoztunk létre, azt egy tömbként kell tovább adni a request-nek.**

Egyedül a C# kódban is említett **Format = DocumentFormat.DF_PDF** megadása kötelező, de a PudoDocumentService.php konstruktora ezt automatikusan elvégzi számunkra.

```
$MassDocumentRequest = new MassDocumentRequest();  
//kikell tölteni a Request-et  
//vonalkódokat tartalmazó tömb  
$MassDocumentRequest->Barcodes = $Barcodes;  
//dokumentum beállításokat tartalmazó tömb  
$MassDocumentRequest->DocumentSettings = $DocumentSettings;  
  
$DocumentClient = new DocumentService($wsdl, $soap_options);  
$MassDocumentResponse = $DocumentClient->GetDocument($MassDocumentRequest);
```

Az így létrehozott példány **GetDocument()** metódusát meg lehet hívni, és visszakapjuk a választ, amiben benne lesz a vonalkódokhoz tartozó dokumentumok **Documents**, a vonalkódonkénti dokumentumok generálása közben keletkezett üzenetek **DocumentGenerationResults** és **Result** kód, amit hibakezelésnél is jól lehet használni, ha esetleg a folyamat valamiért nem hajtott végre. Ne próbáljuk meg elérni a Documents objektumot, ha nem is létezik!

5.2.2.1. Hibakezelés és adatfeldolgozás

Az előbb említett **Result** értéke alapján jól definiálható, hogy kaptunk-e vissza feldolgozható adatot, vagy sem.

```
switch($MassDocumentResponse->GetDocumentResult->Result){  
    case "RES_OK": {  
        //a response sikeres volt, fel lehet dolgozni a legenerált dokumentumokat.  
        //Példa kód lentebb  
        ...  
        break;  
    }  
    case "RES_DATA_MISSING": {  
        echo "Hiányzó adatok miatt nem lehet előállítani dokumentumokat";  
        break;  
    }  
    case "RES_GENERATE_ERROR": {  
        echo "Dokumentum előállítás közben hiba történt";  
        break;  
    }  
    default : {  
        echo $MassDocumentResponse->GetDocumentResult->Result;  
        break;  
    }  
}
```



RES_OK esetén az alább látható módon lehet feldolgozni a dokumentumokat.

```
//Válasz üzenetek a dokumentumokról
foreach($MassDocumentResponse->GetDocumentResult->DocumentGenerationResults->DocumentGenerationResult as $DocumentGenerationResult) {
    print("DocumentName: " . $DocumentGenerationResult->DocumentName . "<br>");
    print("Result: " . $DocumentGenerationResult->Result . "<br>");
    print("ResultMessage: " . $DocumentGenerationResult->ResultMessage . "<br>");
    print("SupplimentJSONData: " . $DocumentGenerationResult->SupplimentJSONData . "<br>");
    print("<br>");
}
//Egyszerűsített dokumentum lemezre írás
foreach($MassDocumentResponse->GetDocumentResult->Documents->DocumentData as $docResponse){
    $PrintDoc = fopen($docResponse->DocumentName.".pdf", "w");
    fwrite($PrintDoc, $docResponse->Document);
    fclose($PrintDoc);
    echo $docResponse->DocumentName.".pdf fájl írása megtörtént<br>";
}
```

Ez a pár sor példa a visszakapott adatok miként történő feldolgozására.

Ezek alapján sikeresen elkészíthető php forrás kód, mely **Valid** csomag vonalkódok alapján visszakapja a kinyomtatható csomag címkéket, készpénz- és csomagátvételi elismervényeket.

Amennyiben a **\$MassDocumentResponse->GetDocumentResponse** nem jönne létre, úgy magával a soapClient hívással lehet probléma.

6. Csomag nyomkövetés (Tracking)

A rendszerbe regisztrált csomagok életútja során bekövetkezett státusz változásokról igény esetén nyomkövetési információkat küldünk. Ehhez az szükséges, hogy szerződött partnerünk a serverén megvalósítson egy Web API (REST) (<https://docs.microsoft.com/en-us/azure/architecture/best-practices/api-design>) fogadó oldalt, ahova HTTP POST hívással, JSON formátumban kiküldjük az adatokat.

Ajánlott az API-t a következő url struktúrában elhelyezni: <https://websitename.com/api/tracking>

Ajánlott https tanúsítvánnyal ellátni a weboldalt, mert így titkosított csatornán tudjuk kiküldeni a tracking információkat, ennek hiányában az adatok harmadik fél birtokába kerüléséért felelősséget nem vállalunk.

A csomagok státusz változásairól körülbelül 10 percenként fogunk információt küldeni. A JSON objektum tartalmazni fogja a partner telephelyéhez tartozó összes csomag elmúlt 10 percben bekövetkezett státuszváltozását, a Pick Pack Pont-os és a Házhoz szállítós csomagokra egyaránt.

Ha a server nem volt elérhető, vagy bármilyen hibás választ kaptunk vissza (nem „success-t”), akkor később megpróbáljuk újraküldeni az adatot. Előfordulhat, hogy ilyenkor az újraküldés előtt sikeresen kiküldünk újabb státuszváltozást is. Ezt partneri oldalon kell kezelni. A TrackingID-val lehet ellenőrizni, hogy már rögzítésre került-e az adott státuszváltozás, a státuszváltozások logikai sorrendjét pedig az EventCreatedTime adja.

6.1. Request

A HTTP request ContentType-ja: „application/json; charset=utf-8”, az adatot HTTP POST hívással küldjük el. A küldött JSON fájl formátuma és a küldött adatok részletes leírása az alábbiakban található. Minden szükséges adatot, státuszt kiküldünk, így partnerünk saját hatáskörben döntheti el, hogy ezek közül melyiket kívánja tárolni, milyen logikával dolgozza fel azokat.

UnifiedParcelTrackingInfo

Típus: Osztály

Leírás:

A tracking információkat magában foglaló osztály.

Mezők:

Típus	Mezőneve	Megjegyzés	Kötelező
string	PartnerCode	Partner (feladó) azonosító	Igen
List<TrackingInfo>	TrackingInfoList	Tracking adatok listája	Igen

TrackingInfo

Típus: Osztály

Leírás:

A TrackingInfoList tartalmazza az összes, csomagstátusz-változásokra vonatkozó információt. A Szolgáltatás oszlopban van megjelölve, hogy az adott mező csak Pick Pack Pont (PPP), csak Házhoz szállítás (HD), vagy mindkettő (PPP+HD) szolgáltatás esetében került-e kitöltésre.

Főbb mezők

Ezek a mezők tartalmazzák a nyomkövetési információk feldolgozásához mindenképpen szükséges adatokat.

Típus	Mezőneve	Megjegyzés	Kötelező	Szolgáltatás
int	TrackingID	A tracking adat egyedi azonosítója. Ezzel lehet ellenőrizni, hogy küldtük-e már ezt az adatot.	Igen	PPP+HD
string	ParcelBarcode	Csomag azonosító, ez a csomag logisztikai azonosítója	Igen	PPP+HD
DateTime	EventCreatedTime	A státuszváltozás időpontja	Igen	PPP+HD
string	DeliveryMode	Szállítási mód, lehetséges értékei: PICKPACKPOINT vagy HOMEDELIVERY	Igen	PPP+HD

Státuszok

Alapvetően kétféle státuszt különböztetünk meg, a Partneri és a Vevői tracking státuszt.

A **Partneri tracking státusz** részletesebb információt ad a csomag állapotáról. Ezen belül van egy Partneri fő státusz(**PartnerMainStatus**) és Partneri al-státusz(**PartnerSubStatus**) ami részletesebben kifejti, hogy mi történik a csomaggal. (Például fő státusz: „Feldolgozás alatt”, al-státusz lehet „raktári feldolgozás” vagy „szállítás alatt”. Ezek a státuszok láthatóak a Partner Portálon is a nyomkövetési információk alatt. <https://portal.pickpackpont.hu/#/tracking>

A **Vevői tracking státusz (CustomerStatus)** a vásárló számára releváns státuszokat tartalmazza. (Például „A csomag megérkezett a PickPack Pontra. Pick Pack Pont megnevezése: xxxx, Pick Pack Pont címe:” Ezeket az információkat láthatja a vásárló a <https://www.pickpackpont.hu/> vagy a <https://www.sprinter.hu/> Csomagkereső menüpontja alatt.



A lehetséges státuszok a **TrackingStatus.xlsx** dokumentumban találhatóak. Vevői tracking esetében ahol NULL szerepel, ott a legutóbbi aktuális státuszt küldjük. A **Sztornó ok (RefuseReason)** Házhozszállítási kiszállításnál ha a vevő nem vette át a csomagot, akkor ennek az okát tartalmazza.

Az egyes státusz információkhoz 2 mező tartozik. Egy **Name** mező, ami a státusz egyedi elnevezése, erre lehet logikát építeni. (Például: WAITING_FOR_PACKAGE, DELIVERED). Valamint egy **DescriptionList** mező, ami magyar és angol nyelven tartalmazza a státuszok szövegezését, amely a felhasználók számára is értelmezhető. Fontos, hogy ezek a leírások idővel változhatnak.

Típus	Mezőneve	Megjegyzés	Kötelező	Szolgáltatás
string	PartnerMainStatusName	Partneri főstátusz neve	Igen	PPP+HD
List<Translation>	PartnerMainStatusDescriptionList	Partneri főstátusz leírása	Igen	PPP+HD
string	PartnerSubStatusName	Partneri alstátusz neve	Nem	PPP+HD
List<Translation>	PartnerSubStatusDescriptionList	Partneri alstátusz leírása	Nem	PPP+HD
string	CustomerStatusName	Vevői státusz neve	Igen	PPP+HD
List<Translation>	CustomerStatusDescriptionList	Vevői státusz leírása	Igen	PPP+HD
string	RefuseReasonName	Sztornó ok neve	Nem	HD
List<Translation>	RefuseReasonDescriptionList	Sztornó ok leírása	Nem	HD

Egyéb azonosítók

A csomagok könnyebb beazonosíthatósága érdekében küldjük a partnerek által ismert egyéb azonosítókat

Típus	Mezőneve	Megjegyzés	Kötelező	Szolgáltatás
string	HomeDeliveryNoteNumber	Fuvarlevél azonosító. Házhoz szállítási csomagoknál az egy címre szállított csomagok azonosítója	Nem	HD
string	ShipmentID	A csomag regisztrációhoz tartozó Shipment azonosítója, amit a háttérrendszer generál. ServiceType-	Nem	PPP+HD



		onként egy azonosító alá kerülnek a csomagok PPP esetén. Házhoz szállítás esetén minden Parcel-hez külön generálódik.		
string	OriginalBarCode	Eredeti vonalkód, amivel a csomagot rögzítették.	Nem	PPP+HD
string	AuthorizationCode	Megrendelés kódja, ami a partner saját csomag azonosítója, vagy a termék számlájának a száma.	Nem	PPP+HD
string	InvoiceNumber	Megszűnt mező. A csomagban lévő számla száma. Szerepét az AuthorizationCode vette át.	Nem	HD

Egyéb információk

Típus	Mezőneve	Megjegyzés	Kötelező	Szolgáltatás
string	CustomerName	Vásárló neve	Nem	PPP+HD
DateTime?	ExpectedArrivalFrom	Kézbesítés várható idejének eleje	Nem	PPP+HD
DateTime?	ExpectedArrivalTo	Kézbesítés várható idejének vége	Nem	PPP+HD
string	DestinationLocationCode	Cél PickPackPont azonosítója	Nem	PPP
string	DestinationLocationName	Cél PickPackPont neve	Nem	PPP
string	DestinationLocationAddress	Cél PickPackPont címe	Nem	PPP
string	InternationalDeliveryPartnerName	Nemzetközi szállító cég neve, pl FanCourier	Nem	HD
Decimal?	MeasuredWeight	A csomag mért súlya kg-ban.	Nem	HD

Translation

Típus: Osztály

Leírás:

Magyarul és angolul tartalmazza a státuszok felhasználó számára értelmezhető leírását.

Mezők:

Típus	Mezőneve	Megjegyzés	Kötelező
string	LanguageName	Nyelv megnevezése, HU vagy EN.	Igen
string	TranslatedText	Lefordított leírás	Igen

6.2. Response

A válasz szabványos HttpResponse, a megfelelő HttpStatusCode kitöltésével, sikeres rögzítés esetén **HttpStatusCode 200 OK**-val és a „success” content-tel. A success válasz visszaadása kötelező, mivel innen tudjuk, hogy a mentés sikeresen megtörtént!

6.2.1. Hibakódok

A válaszban az alábbi HttpStatusCode-ok fordulhatnak elő. Bővebben:

https://en.wikipedia.org/wiki/List_of_HTTP_status_codes

Status code	Status name	Content	Megjegyzés
200	OK	success	Sikeres rögzítés
404	Not found	No HTTP resource was found	A megadott URL nem létezik
405	Method Not Allowed	Request method must be POST!	Nem POST hívás érkezett
415	Unsupported Media Type	Content type must be: application/json; charset=utf-8. Request content type was:	A kérés tartalma nem megfelelő formátumú
400	Bad Request	The request is invalid.	A kapott objektum nem megfelelő formátumú, emiatt nem lehet feldolgozni. Más struktúrájú objektumot kaptunk, mint amit vártunk.
500	Internal Server Error	Error occured! ErrorMessage:	Nem várt hiba történt.



6.3. Forráskód demó

A tracking információk szerver oldali fogadására példakódot mellékelünk C# és PHP nyelveken.

6.3.1. C# kód

A Web API fogadó oldalt megvalósító .Net forrás: a mellékletben található az interface-t leíró dokumentum mellett.

csharp/Tracking/PUDOParcelTracking.sln

Az sln Visual Studio 2015 megnyitásával betölti a példa kódot.

A demóban egy ASP.NET Web API 2.0 projekt található. A projekt IIS vagy IIS express alatt futtatható, a szükséges NuGet Package-eket Restore-olni kell, ha automatikusan nem töltené le a Visual Studio.

A Controllers\TrackingController.cs fájlban található az Api-hoz tartozó Controller és a Post hívást fogadó metódus. Paraméterben egy UnifiedParcelTrackingInfo objektumot vár, amely a fentebb leírt adatstruktúrában épül fel és a Models mappa alatt található.

A példakódban látható módon ellenőrizni kell a kérés helyességét, hogy hiba esetén megfelelő választ kapjunk vissza!

A válaszban mindig ezt kell visszaadni:

```
return Request.CreateResponse(HttpStatusCode.OK, "success");
```

6.3.2. PHP kód

A Web API fogadó oldalt megvalósító PHP forrás: a mellékletben található az interface-t leíró dokumentum mellett.

php/Tracking/TrackingController.php

A demó kód egy HTTP post kérést fogad, ellenőrzi ennek helyességét.

Ha valóban POST kérés érkezett 'application/json; charset=utf-8' ContentType-al, akkor a küldött adatot JSON-ként értelmezzük:

```
//Receive the RAW post data.  
$request = trim(file_get_contents("php://input"));
```

```
//Attempt to decode the incoming RAW post data from JSON.  
$contentObjectFromJson = json_decode($request);
```

A JsonMapper osztályt (<https://github.com/cweiske/jsonmapper>) arra használjuk, hogy a kapott JSON objektumot UnifiedParcelTrackingInfo típusra át tudjuk alakítani és validáljuk, hogy a kötelező adatok ki vannak-e töltve, megfelelő formátumú-e az adat. Ha sikertelen a validáció, akkor hibát ad vissza.

Ezt követően már tetszés szerint fel lehet dolgozni a kapott JSON objektumot vagy a típusos `UnifiedParcelTrackingInfo` objektumot.

A válaszban mindig HTTP 200 OK válasszal a „success” üzenetet kell visszaadni.

```
//At the end return success with HTTP status 200
echo "success";
```

6.3.3. Tesztelés

A demó kódokat tetszőleges REST API-t hívni tudó alkalmazással lehet tesztelni- pl Soap UI (<https://www.soapui.org/>). Tesztelésnél fontos, hogy POST metódussal küldjük be az adatokat és a Header ContentType „application/json” legyen!

Példa JSON adatok: egy PPP és egy HD regisztrált csomag (a hatékony helykihasználás érdekében másoltuk, de webes tool-okkal megnézhető formázva is- (PI <https://jsonformatter.org/>):

```
{
  "PartnerCode": "0000001000",
  "TrackingInfoList": [
    {
      "TrackingID": 1,
      "ParcelBarcode": "tracking_test_PP",
      "HomeDeliveryNoteNumber": null,
      "ShipmentID": "SP201802270000000012",
      "OriginalBarCode": "tracking_test_PPP",
      "AuthorizationCode": "auth_tracking_test_PPP",
      "InvoiceNumber": null,
      "EventCreatedTime": "2018-03-27T09:06:03.5249597+02:00",
      "DeliveryMode": "PICKPACKPOINT",
      "PartnerMainStatusName": "WAITING_FOR_PACKAGE",
      "PartnerMainStatusDescriptionList": [
        {
          "LanguageName": "HU",
          "TranslatedText": "A csomag beérkezésére várunk"
        },
        {
          "LanguageName": "EN",
          "TranslatedText": "Waiting for the parcel to arrive"
        }
      ],
      "PartnerSubStatusName": null,
      "PartnerSubStatusDescriptionList": null,
      "CustomerStatusName": "WAITING_FOR_PACKAGE_SENDER",
      "CustomerStatusDescriptionList": [
        {
          "LanguageName": "HU",
          "TranslatedText": "A csomagot még nem adta át a Feladó szállításra. További információért kérjük keresse a Feladót (webáruházat)."
        },
        {
          "LanguageName": "EN",
          "TranslatedText": "The sender haven't handed over the parcel yet. For more information, please call the sender (webshop)."
        }
      ],
      "RefuseReasonName": null,
      "RefuseReasonDescriptionList": null,
      "CustomerName": "Teszt Elek",
      "ExpectedArrivalFrom": "2018-03-28T09:06:03.5259473+02:00",
      "ExpectedArrivalTo": "2018-03-29T09:06:03.5259473+02:00",
      "DestinationLocationCode": "0000101005",
      "DestinationLocationName": "Teszt Shop",
      "DestinationLocationAddress": "1164, Budapest, Szabadszabó utca 1.",
      "InternationalDeliveryPartnerName": null,
      "MeasuredWeight": null,
      "TrackingID": 2,
      "ParcelBarcode": "tracking_test_HD",
      "HomeDeliveryNoteNumber": "tracking_test_HD",
      "ShipmentID": "SP201802270000000013",
      "OriginalBarCode": "tracking_test_HD",
      "AuthorizationCode": "auth_tracking_test_HD",
      "InvoiceNumber": null,
      "EventCreatedTime": "2018-03-27T09:06:03.5269488+02:00",
      "DeliveryMode": "HOMEDELIVERY",
      "PartnerMainStatusName": "WAITING_FOR_PACKAGE",
      "PartnerMainStatusDescriptionList": [
        {
          "LanguageName": "HU",
          "TranslatedText": "A csomag beérkezésére várunk"
        },
        {
          "LanguageName": "EN",
          "TranslatedText": "Waiting for the parcel to arrive"
        }
      ],
      "PartnerSubStatusName": null,
      "PartnerSubStatusDescriptionList": null,
      "CustomerStatusName": "SPRINTER_WAITING_FOR_PACKAGE_SENDER",
      "CustomerStatusDescriptionList": [
        {
          "LanguageName": "HU",
          "TranslatedText": "A csomagot még nem adta át a Feladó szállításra. További információért kérjük keresse a Feladót (webáruházat)."
        },
        {
          "LanguageName": "EN",
          "TranslatedText": "The sender haven't handed over the parcel yet. For more information, please call the sender"
        }
      ]
    }
  ]
}
```



(webshop)."}}, "RefuseReasonName": null, "RefuseReasonDescriptionList": null, "CustomerName": "Teszt Elek", "ExpectedArrivalFrom": "2018-03-28T09:06:03.5269488+02:00", "ExpectedArrivalTo": "2018-03-29T09:06:03.5269488+02:00", "DestinationLocationCode": null, "DestinationLocationName": null, "DestinationLocationAddress": null, "InternationalDeliveryPartnerName": null, "MeasuredWeight": 3.14}}}