

Külső webshoppal való kommunikáció és fontosabb infók

Webshop egyedi szinkron:

Egyedi szinkronizálás a termékek evir oldalon megtalálható, és a webshopban egy (ritka esetben több) api hívással módosítható, feltölthető. Az egyedi szinkronizáció szolgál a termékek **feltöltésére** is.

- A szinkront futtatja a my[webshop].pm: egy dok.pm alapú objektum, mely a [webshop]_common.pm ből összerakott \$self→{Data}→{Mezok} -el dolgozik.
 - Ezeket a mezóket a következő módon lehet testreszabni, két fő csoportra lehet osztani:
 - Egyértelmű az evir oldalon az adat, és egyértelműen van helye a webshop oldalon.
 - például:
 - megnevezés,
 - ar,
 - készlet
 - kategória
 - tulajdonságok
 - kapcsolódó és helyettesítő termékek
 - az {options} értékei:
 - 0: nem visszük át a webshopba
 - default: Átvisszük a webshopba
 - eVIR oldaláról nem egyértelmű az adat beazonosítása, vagy nem létezik ilyen adat, akkor megengedjük a termék tulajdonság feltöltését is.
 - például
 - leírás mezők,
 - súly
 - megjegyzés mezők
 - slug
 - az option értékei:
 - 0: nem visszük át a webshopba
 - default: Valami általunk választott cikkparam érték
 - pl Bruttó súly
 - más általunk választott cikkparam érték,
 - pl Netto súly
 - Termék tulajdonságok felsorolva.
 - Az árat és a készletet külön kezeljük, mert volt, mikor nem engedték, hogy egyedi szinkronizálásban szerepeljenek. Ez viszont XXX.
 - torzs_webshop::sync_product:
 - abstract
 - lista az egyedi szinkronizálásokról.
 - Ezen listából lehet futtatni is őket, vagy módosítani.
 - Összes termék szinkronizálása azt jelenti, hogy a last_sync a legkorábbi időpontra lesz állítva, a last_cikkszám pedig üres string lesz.
 - [webshop]_sync tablában vannak a fej adatok tarolva
 - last_sync és a last_cikkszám a batchesitest szolgálják, ahol a last_sync a last_cikkszám modify_time-ja.
 - id: azonosítja a szinkront
 - [webshop]_sync_mezok tablában vannak, hogy mely termék rekordokat kell szinkronizálni, és milyen értékkel.
 - torzs_webshop::sync_setup:
 - Form, melyet az egyedi szinkron listából lehet elérni. Egy egyedi szinkron

letrehozasa/modositasara szolgal.

- itt lehet beallitani a [webshop]_sync_mezok erteket valamint a sync nevet.

Webshop folyamata:

Altalanos info:

- torzs_webshop::common -t haszalom a cikkeknek az evirre vonatkozó adatok összeszedesehez.
- torzs_webshop::webshop_log*.pm logolashoz es log listához.
- 'webshop_log_fej' es 'webshop_log_tetel' tablakban van a log
 - webshop_log_fej:muvelet: pl termekek szinkronizalasa
 - webshop_log_tetel: amikkel a muvelet vegrehajtott. pl termekek
- elib::error_notifier hasznalata a notifiolashoz.
 - jelenleg emailt kuldok ki
- [webshop]_variables tablaban vannak olyan ertekek, melyekre szuksegunk van ket api muvelet kozott, de nem annyira nagyok, hogy teljes tablat kapjanak
 - mint az utolso rendeles id, ar szinkronizalasa

Termek szinkronizalasa folyamata:

- A lefutott szinkronizalasnak(mely lehet egyedi de lehet ar/keszlet is) vesszuk az last_sync es a last_cikkszam rekordjait
 - last_sync: utoljara felszinkronizalt termek modositasi timestamp-je
 - last_cikkszam: utoljara felszinkronizalt termek cikkszama.
- ezekkel szurunk a termekekre, hogy csak azokat szinkronizaljuk, mely a last_sync ota valtozott. Ha a modify_time=last_sync akkor a last_cikkszammal hasonlitjuk össze.
- (altalaban) 50 darab cikkre szurunk, es ezt egy darab api hivással frissitjuk miutan összeszedtuk a szukseges adatokat.
 - oka: woocommerce: Kiszamoltuk, hogy noha 100 a batch merete, de 50-el fut le emberi idon belül.
- Mielott az api hivast megtesszuk, frissitjuk a last_sync es last_cikkszam-ot, hogy ne akadjon be a szinkronizalasa hiba eseten.
- Ez utan megnezzuk, hogy van-e meg idonk lefuttatni egy ujab hivast (az elozo futasi idot felhasznalva a becsleshez).
 - Ha van akkor megcsinaljuk
 - Ha nincs akkor a hatterben fog futni tovabb, batchjobkent.
- Ar es keszlet szinkronizalaskor nem toltunk fol uj cikket, csak a meglevoket frissitjuk a gyorsasag miatt.
- keszlet szinkronizalaskor nem a modositasi datumot vesszuk alapul a cikkek szuresere, hanem, hogy last_sync ota megjelent-e az arslot_log tablaban bejegyzes raktarkeszlet valtozasra.

sync_product_by_id megjegyzesek

— [Csesznák Tamás](#) 2022/08/26 11:32 A megjegyzések nagyrésze lehet, hogy a source_browserben is megtalálható, és sokat átlehet oda költöztetni.

- a \$id az egyedi szinkronizalások evires id-je.
- sub _sync_products_init: Inicializálja azokat a {Mezok}-et amiket szinkronizálni lehet
- set_value hogy mely mezoket akarom szinkronizálni
 - ettől függően metadaadatok lekerese
 - Vagy eVIR oldalon kapcsolotablal
 - Vagy webshop api hivással
- while ciklusban amíg lehet:
 - sub calc_product_sync_infos:

- Elsődlegesen eVIR oldali cikk adatok összeszedése, a következő 50 darabra.
- Másodlagosan (shoppoktól eltérően ez változhat), itt állapítom meg, hogy mely cikk van feltöltve, és mely nincsen.
- sub calc_product_sync_infos_ar:
 - gyors megoldás a sale arkatégória megszerzéséhez
 - Ronda, és javítani kéne, költöztetni a torzs_webshop::common-ba
- sub sync_products_felvesz
 - paraméterek:
 - do_update⇒frissitunk
 - do_create⇒letrehozunk
 - sync_name⇒'szinkron neve'
- végigmegy a cikkeken, és megnezi, hogy az adott cikket update-elni vagy create-elni kell.
 - sub sync_update_product:
 - Készít egy rekordot, amit oda lehet adni az apinak.
 - shopify esetén itt végzi el a szinkronizálást is.
- visszatér a sikerek számával (jó esetben ez az elvárt működés).

keszlet szinkronizacio:

Keszleteknel annyi változás van, hogy a termékeket nem a modify datum alapján, hanem az aruslot log-ban történt változás alapján szedem össze.

Webshop specifikus infok:

woocommerce:

tulajdonság kategória és címkéknek van metaadata amit fel kell tölteni szinkron előtt.

vannak kapcsolótáblák. A woocommerce_common.pm-ben vannak az utasítások hogy berakjuk a táblába a terméket.

minden metaadatnak, a media adatoknak és a termékeknek van táblája.

A képek feltöltését a wordpress intézi

A képeket nem lehet webshop oldali batchként feltölteni, csak 1-esével.

Shoprenter:

Mindig lekerjük a shopból a termékeket. Nincs kapcsolótábla

kapcsolótábla van:

képek

A nyelv adatokat lekerjük minden hívásnál, hogy a description-nek megtudjuk mondani a nyelvet .

Api kisokos:

A maximálisan indítható requestek limitje 3/alkalmazás/bolt/másodperc.

Outer ID-t nem használunk, mert nem támogatott, helyette az extended resource-ot használjuk.

Minden olyan mező, ami leírást/szöveget tartalmaz 2 részből áll: 1 fej részből, és egy description részből. Például a text tulajdonságok, vagy a termék megnevezése.

Az id mezők base64 kódolással, kibontva ezt kapjuk:

pl: productCategory-product_id=534&category_id=146

UNAS:

XML-ben kommunálunk.

unas_data.pm alakítja pxml adatszerkezeté.

Nincs adat, amit külön kellene feltölteni.

max 20 darab sikertelen hívás, és max 128 mb.

augusztus 9 óta a megrendelés státusz is lekérdezhető API-n keresztül

Shopify:

GraphQL-t is használunk pl készletnél.

nincs shop oldali batchelt hívás

Nincsenek termék tulajdonságok:

Csak options van, amiből 3 választható

— [Csesznák Tamás](#) 2022/08/26 11:12

From:

<https://www.doc.evir.hu/> - **eVIR tudásbázis**

Permanent link:

https://www.doc.evir.hu/doku.php/spec:kulso_webshop

Last update: **2022/08/26 09:45**

