

Külső webshoppal való kommunikáció és fontosabb infók

Webshop egyedi szinkron:

Egyedi szinkronizálás a termékek evir oldalán 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 mezőket a következő módon lehet testreszabni, két fő csoportra lehet osztani:
 - Egyértelmű az evir oldalán az adat, és egyértelműen van helye a webshop oldalán.
 - 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 tárolva
 - `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 létrehozása/modosítására szolgál.
 - itt lehet beállítani a [webshop]_sync_mezok értéket valamint a sync nevet.

Webshop folyamata:

Általános info:

- torzs_webshop::common -t haszalom a cikkeknek az evirre vonatkozó adatok összeszedéséhez.
- torzs_webshop::webshop_log*.pm logoláshoz és log listához.
- 'webshop_log_fej' és 'webshop_log_tetel' tablakban van a log
 - webshop_log_fej:muvelet: pl termékek szinkronizálása
 - webshop_log_tetel: amikkel a művelet végrehajtott. pl termékek
- elib::error_notifier használata a notifikáláshoz.
 - jelenleg emailt küldök ki
- [webshop]_variables tablában vannak olyan értékek, melyekre szükségünk van két API művelet között, de nem annyira nagyok, hogy teljes tablát kapjanak
 - mint az utolsó rendelés ID, az szinkronizálás

Termék szinkronizálás folyamata:

- A lefuttatott szinkronizálásnak (mely lehet egyedi de lehet ár/készlet is) vesszük az last_sync és a last_cikkszám rekordjait
 - last_sync: utóljára felszinkronizált termék módosítási timestamp-je
 - last_cikkszám: utóljára felszinkronizált termék cikkszám.
- ezekkel szűrünk a termékekre, hogy csak azokat szinkronizáljuk, mely a last_sync óta változott. Ha a modify_time=last_sync akkor a last_cikkszámmal hasonlítjuk össze.
- (általában) 50 darab cikkre szűrünk, és ezt egy darab API hívással frissítjük miután összeszedtük a szükséges adatokat.
 - oka: woocommerce: Kiszámoltuk, hogy noha 100 a batch mérete, de 50-el fut le emberi időn belül.
- Mielőtt az API hívást megtegyük, frissítjük a last_sync és last_cikkszám-ot, hogy ne adjon be a szinkronizálás hiba esetén.
- Ez után megnezzük, hogy van-e még időnk lefuttatni egy újabb hívást (az előző futási időt felhasználva a becsléshez).
 - Ha van akkor megcsináljuk
 - Ha nincs akkor a háttérben fog futni tovább, batchjobként.
- Ár és készlet szinkronizálásakor nem töltünk föl új cikket, csak a meglévőket frissítjük a gyorsaság miatt.
- készlet szinkronizálásakor nem a módosítási dátumot vesszük alapul a cikkek szűrésére, hanem, hogy last_sync óta megjelent-e az arslot_log tablában bejegyzés raktárkészlet változásra.

sync_product_by_id megjegyzések

— [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 szinkronizálások evires ID-je.
- sub _sync_products_init: Inicializálja azokat a {Mezők}-et amiket szinkronizálni lehet

- `set_value` hogy mely mezőket akarom szinkronizálni
 - ettől függetlenül metadatok lekérése
 - Vagy eVIR oldalon kapcsolótáblával
 - Vagy webshop API hívással
- `while` ciklusban amíg lehet:
 - `sub calc_product_sync_infos`:
 - Elsődlegesen eVIR oldali cikk adatok összeszedése, a következő 50 darabra.
 - Másodlagosan (shopoktól eltérően ez változhat), itt állapítom meg, hogy mely cikk van feltöltve, és mely nincsen.
 - `sub calc_product_sync_infos_ar`:
 - gyors megoldás a sale kategória megszerzéséhez
 - Ronda, és javítani kéne, költöztetni a `torzs_webshop::common-ba`
 - `sub sync_products_felvesz`
 - paraméterek:
 - `do_update`⇒frissítünk
 - `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 API-nak.
 - Shopify esetén itt végzi el a szinkronizálást is.
 - visszatér a sikerek számával (jó esetben ez az elvárt működés).

készlet szinkronizáció:

Készleteknél annyi változás van, hogy a termékeket nem a modify dátum alapján, hanem az árslót log-ban történő változás alapján szedem össze.

Webshop specifikus infók:

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 média adatoknak és a termékeknek van táblája.  
A képek feltöltését a WordPress intézi  
A képeket nem lehet webshop oldali batchként feltölteni, csak 1-esevel.
```

Shopreenter:

```
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 mezok base64 kodolasuaki, kibontva ezt kapjuk:

pl: productCategory-product_id=534&category_id=146

UNAS:

XML-ben kommunikálunk.

unas_data.pm alakítja pxml adatszerkezetet.

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://doc.evir.hu/> - eVIR tudásbázis

Permanent link:

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

Last update: **2022/08/26 11:45**

