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 mezoket a kovetkezo modon lehet testreszabni, két fő csoportra lehet osztani:
        - Egyertelmű az evir oldalon az adat, és egyértelműen van helye a webshop oldalon.
            - peldaul:
                - megnevezés,
                - ar,
                - készlet
                - kategoria
                - tulajdonságok
                - kapcsolódó és helyettesítő termékek
            - az `{options}` értékei:
                - 0: nem visszük at a webshopba
                - default: Átvisszük a webshopba
        - eVIR oldalaról 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 ertekei:
            - 0: nem visszuk at a webshopba
            - default: Valami általunk választott cikkparam érték
                - pl Bruttó súly
            - mas általunk választott cikkparam érték,
                - pl Netto súly
            - Termék tulajdonságok felsorolva.
    - Az árat és a készletet külön kezeljük, mert volt, mikor nem engedtük, 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_cikkszam` pedig üres string lesz.
        - [webshop]_sync tablaban vannak a fej adatok tarolva
            - last_sync es a last_cikkszam a batchesitest szolgaljak, ahol a last_sync a last_cikkszam modify_time-ja.
            - id: azonositja a szinkront
        - [webshop]_sync_mezok tablaban vannak, hogy mely termek rekordokat kell szinkronizalni, es milyen ertekkel.
- torzs_webshop::sync_setup:
    - Form, melyet az egyedi szinkron listabol lehet elerni. 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 vonatkozo adatok osszeszedesehez.
- torzs_webshop::webshop_log*.pm logolashoz es log listahoz.
- 'webshop_log_fej' es 'webshop_log_tetel' tablakban van a log
    - webshop_log_fej:muvelet: pl termekek szinkronizalasa
    - webshop_log_tetel: amikkel a muvelet vegrehajtodott. pl termekek
- elib::error_notifyer hasznalata a notifyolashoz.
    - 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 szinkronizalas

Termek szinkronizalas folyamatas:

- A lefuttatott 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 ossze.
- (altalaban) 50 darab cikkre szurunk, es ezt egy darab api hivassal frissitjuk miutan osszeszedtuk a szukseges adatokat.
    - oka: woocommerce: Kiszamoltuk, hogy noha 100 a batch merete, de 50-el fut le emberi idon belul.
- Mielott az api hivast megtesszuk, frissitjuk a last_sync es last_cikkszam-ot, hogy ne akadjon be a szinkronizalas hiba eseten.
- Ez utan megnezzuk, hogy van-e meg idonk lefuttatni egy ujabb hivast (az elozo futasi idot felhasznalva a becsleshez).
    - Ha van akkor megcsinaljuk
    - Ha nincs akkor a hatterben fog futni tovabb, batchjobkent.
- Ar es keszlet szinkronizalasakor 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 aruslot_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 szinkronizalasok evires id-je.
- `sub _sync_products_init`: Inicializalja azokat a {Mezok}-et amiket szinkronizalni lehet
- set_value hogy mely mezoket akarom szinkronizalni
    - ettol fuggoen metadaadatok lekerese
        - Vagy eVIR oldalon kapcsolotablabol
        - Vagy webshop api hivassal
- while ciklusban amig 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 megoldas a sale arkategoria megszerzesehez
        - Ronda, és javítani kéne, költöztetni a torzs_webshop::common-ba
    - sub sync_products_felvesz
        - parameterek:
            - do_update⇒frissitunk
            - do_create⇒letrehozunk
            - sync_name⇒'szinkron neve'
    - vegigmegy a cikkeken, es megnezi, hogy az adott cikket update-elni vagy create-elni kell.
        - sub sync_update_product:
            - Keszit egy rekordot, amit oda lehet adni az apinak.
            - shopify esetén itt végzi el a szinkronizálást is.
    - visszater a sikerek szamaval (jó esetben ez az elvárt működés).

keszlet szinkronizacio:

Keszleteknel annyi valtozas van, hogy a termekeket nem a modify datum alapjan, hanem az aruslot log-ban torteno valtozas alapjan szedem ossze.

Webshop specifikus infok:

woocommerce:

```
  tulajdonsag kategoria es cimkenek van metaadata amit fel kell toltenni szinkron
elott.
  vannak kapcsolotablak. A woocommerce_common.pm-ben vannak az utasitasok hogy
berakjuk a tablaba a termeket.
  minden metaadatnak, a media adatoknak es a termekeknek van tablaja.
  A kepek feltolteset a wordpress intezi
      A kepeket nem lehet webshop oldali batchkent feltolteni, csak 1-esevel.
```

Shoprenter:

```
  Mindig lekerjuk a shopbol a termekeket. Nincs kapcsolotabla
  kapcsolotabla van:
  kepek
  A nyelv adatokat lekerjuk minden hivasnal, 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 komunikalunk.
 unas_data.pm alakitja pxml adatszerkezette.
Nincs adat, amit külön kellene feltölteni.
max 20 darab sikertelen hivas, es max 128 mb.
agusztus 9 ota a megrendelés státusz is lekérdezhető apin keresztül
```

Shopify:

```
graphQL-t is hasznalunk pl keszletnel.
nincs shop oldali batchelt hivas
Nincsenek termék tuladjonságok:
  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**