



Comifar listino specifiche tecniche

Introduzione

Scopo del software è quello di scaricare il listino completo ed i suoi aggiornamenti dal webservice fornito da Comifar.

I dati saranno caricati all'interno di una tabella chiamata `comifar_listino`.

Lo script controllerà se la tabella esiste. In caso contrario girerà in modalità RESET come descritto in dettaglio nel seguito.

Lo script lavorerà sia da http che da CLI per essere configurato come cronjob.

Per i dati di configurazione è stato sviluppato un modulo chiamato `Intraweb/Comifar`.

Soap client ed importazione dati

Un client soap scaricherà i dati forniti dal web service Comifar chiamato "listino". Questo servizio accetta i seguenti parametri:

`listino (myCLIENTE, myPWD, myPWD2, myARGS, myTIPO) as String`

Se la richiesta fallisce per qualche motivo verrà inviata una mail di notifica all'indirizzo configurato come debug email nel modulo.

Di seguito una chiamata di esempio con l'array di configurazione fornito dal modulo Comifar per Magento in modalità RESET:

```
/* inizializza il client soap con i parametri del WSDL */
```

```
$options = array(  
    'soap_version'=>SOAP_1_2,
```

```

        'exceptions'=>true,
        'trace'=>1,
        'cache_wsdl'=>WSDL_CACHE_NONE,
        "connection_timeout" => 120
    );

```

```

$urlws = $configArray['url'];
$client = new SoapClient($urlws, $options);
$myCLIENTE = (string)$configArray['codcliente'];
$myPWD = (string)$configArray['pwdaz'];
$myPWD2= (string)$configArray['pwd'];
$myARGS = "";
$myTIPO = "RESET";
$response = $client->listino($myCLIENTE, $myPWD, $myPWD2, $myARGS, $myTIPO);

```

\$myTIPO può essere valorizzato come “RESET” o “AGG”, default è AGG

In modalità RESET (la tabella comifar_listino non esiste) avremo questo response:

```

<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:SOAP-ENC="http://schemas.xmlsoap.org/soap/encoding/"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:ns="urn:soapcomifar">
    <SOAP-ENV:Header/>
    <SOAP-ENV:Body SOAP-ENV:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/">
        <ns:listinoResponse>
            <response><!
[CDATA[<listino><info><esito_sigla>OK.RESET</esito_sigla><esito_desc>RICHIESTA
ELABORATA</esito_desc><continua>N</continua><articoli>0</articoli><note></note><progres
sivo>5289</progressivo></info></listino>]]></response>
            </ns:listinoResponse>
        </SOAP-ENV:Body>
    </SOAP-ENV:Envelope>

```

dovremo controllare il tag esito_sigla se è valorizzato con OK.RESET

Se esito_sigla non è valorizzato con OK.RESET il modulo invierà un'email di alert all'indirizzo

inserito nella configurazione.

Se esito_sigla è OK.RESET lo script prosegue con chiamate in modalità AGG fino a che il tag “continua” è valorizzato a S. Lo script si fermerà quanto il tag “continua” sarà uguale a N.

Esempio di response ad una chiamata “AGG”:

<response>

<![CDATA[<listino><info><esito_sigla>OK.AGG</esito_sigla><esito_descr>RICHIESTA ELABORATA. Articoli inviati: 1000 => INVIATO progressivo #5290</esito_descr><continua>S</continua><articoli>1000</articoli><note></note><progressivo>5290</progressivo></info><dati><!

[CDATA[H4sIAAAAAAAAAA81924pIS3LkDyVU3C9QnPf5jAa9DIyEZtQI5u8nPDYNFVnhbrs8 bZ2cg1DrVKfKLWPFxS/m5j//8X/++dfPf/+f//Fff6UyQo4/f+x/+flf//zrf/z8

....

JE281An+H0D2CvWNFAIA]]]]><![CDATA[</dati></listino>]]></response>

come detto controlleremo che esito_sigla sia OK.AGG e continua sia S.

All'interno del tag “dati” avremo il pacchetto dati codificato in base64.

Decodificando, salvandolo all'interno di un file in formato gz e scompattando, avremo un nuovo XML con questa struttura.

<art>

<mins>30860136</mins>

<st>I</st>

<vda>26/01/2016</vda>

<va>31/12/2020</va>

<ls>

<qty>1</qty>

<pn>2,15</pn>

</ls>

<ls>

<qty>3</qty>

<pn>1,31</pn>

</ls>

<ls>

<qty>10</qty>

<pn>1,10</pn>

</ls>

</art>

<art>

<mins>30861090</mins>

<st>I</st>

<vda>01/08/2015</vda>

<va>31/12/2020</va>

<ls>

<qty>1</qty>

<pn>2,07</pn>

</ls>

<ls>

<qty>2</qty>

<pn>1,61</pn>

</ls>

</art>

<art>

<mins>30865036</mins>

<st>I</st>

<vda>01/08/2015</vda>

<va>31/12/2020</va>

<ls>

<qty>1</qty>

<pn>1,83</pn>

</ls>

<ls>

<qty>2</qty>

<pn>1,15</pn>

</ls>

</art>

....

che a sua volta sarà caricato all'interno della tabella comifar_listino.

Il nomi dei file seguono questa convenzione:

dirpath (da config) . ”/” . <progressivo> . gz

dirpath (da config) . ”/” . <progressivo> . Xml

Lo script andrà avanti con le chiamate fino a ce “continua” è uguale a S, ovvero fino a che non avremo il pacchetto finale in cui “continua” uguale a N.

Struttura della tabella comifar_listino e conversione dei dati

La tabella comifar_listino avrà praticamente gli stessi campi del file XML appena visto,

| mins | st | vda | va | price | tier_price |

mins e st hanno gli stessi valori inseriti nel file XML.

Vda e va devono essere convertiti al formato data di mysql (YYYY-mm-aa)

Il primo tag <ls> (qty = 1) fornirà il prezzo standard su magento. I successivi tag <ls> forniranno i tier price e saranno usati per ricavare la stringa dei tier secondo la seguente formula:

"qty_threshold1:price1;qty_threshold2:price2;qty_threshold3:price3"

nell'esempio riportato, il primo articolo con mins (sku) 30860136, inseriremo 2.15 (sostituendo la virgola con il punto) all'interno del campo prezzo, mentre creeremo la stringa per il tier price come segue

“3:1.31;10:1.10”

Export in formato csv e aggiornamento tabella farmadati

Un altro script si occuperà di leggere i dati dalla tabella comifar_listino e creare un file csv. Solo i record contrassegnati dallo status st = I verranno esportati nel file csv, che avrà questa struttura:

sku = mins

price = price

status = 1 se st è I, 0 se st è A

tier_price:_all_ = tier_price

il terzo campo sarà inserito nel csv solo se nella configurazione del modulo comifar è abilitato il campo “tier price”.

Lo stesso script aggiornerà lo status della tabella farmadati ad 1 per tutti gli sku che hanno comifar_listino.st = I, mentre lo status sarà messo a 0 per gli sku che hanno comifar_listino.st = A.

Il csv sarà creato nella dir configurata come finalpath all'interno del modulo comifar.

Cronjob

Il funzionamento dei cronjob è strettamente connesso con quelli di farmadati. In particolare i cronjob di farmadati si occupano di popolare la tabella farmadati con tutti i dati disponibili sulla loro piattaforma. I cronjob di comifar si occupano di produrre tabella e csv comifar e di aggiornare lo status della tabella farmadati. In seguito altri cronjob di farmadati vengono eseguiti per il caricamento dei dati sul database magento ed infine l'ultimo cronjob di Comifar si occupa di aggiornare i prezzi dei prodotti caricati su magento.