AUTOMATIC PERIODIC INCREASE CREDIT LIMIT

Document created on 2014-10-23
Nicolas Bondier

[pdf][doc][htm]

Contents

Introduction

This document describes the script used to periodically increase the credit limit of the customers. It is on slave server in folder `/home/porta-admin/141022-periodic-increase-credit-limit`.

Script documentation

CODE	COMMENT
`#!/usr/bin/env perl` `use strict;` `use warnings;` `use DBI;` `use SOAP::Lite` `# +trace=>'debug'` `;` `use File::Spec::Functions qw(rel2abs);` `use File::Basename;` `use MIME::Lite;` `use Config::IniFiles;` `use POSIX qw(strftime);` `use Data::Dumper;` `use File::stat;` `use Archive::Tar;`	Libraries the script uses.
`my $soap_login = '';` `my $soap_password = '';` `my $cfg = Config::IniFiles->new( -file => "/home/porta-admin/141022-soap-credentials/soap.ini" );` `if ( $cfg->val( 'admin', 'login' ) && $cfg->val( 'admin', 'password' ) ){` `  $soap_login = $cfg->val( 'admin', 'login' );` `  $soap_password = $cfg->val( 'admin', 'password' );` `} else {` `  exit;` `}`	Getting SOAP credentials from configuration file

`` my $dirname = dirname(rel2abs($0)); chdir($dirname); my $log_path = $dirname . '/log/'; my $logfile = "periodic-increase-credit-limit.log"; my $logfile_archive = strftime "%Y-%m-%d_$logfile", localtime; my $logfile_zip = "$logfile_archive.tar.gz"; $logfile = $log_path . "/" . $logfile; $logfile_archive = $log_path . "/" . $logfile_archive; $logfile_zip = $log_path . "/" . $logfile_zip; if (-e $logfile){ my $last_modification_day = sprintf ("%d", stat($logfile)->mtime / (60*60*24)); my $today = sprintf("%d", time / (60*60*24)); if ($last_modification_day != $today && stat($logfile)->size > 0){ print "\nCompressing old file ..."; rename $logfile, $logfile_zip; my $tar = Archive::Tar->new; $tar->write($logfile_zip, COMPRESS_GZIP); print "OK"; } } print "\n"; open (LOGFILE, '>>' .$logfile); ```	Creation of a daily rotating log. The old logs are compressed and stored by date.
`` my $host = 'xxxxxxxxxxxxxxxxxx'; my $port = '3306'; my $database = 'porta-billing'; my $login = 'xxxx'; my $password = ''; my $dbh = DBI->connect("DBI:mysql:database=$database;host=$host;port=$port", $login, $password) or die "Connection impossible à la base de données $database !\n $! \n $@\n$DBI::errstr"; ```	Connection to the master database.

```perl
# SOAP CONNECTION TO NEW BILLING
binmode(STDOUT, ':utf8');
my $proxy_host = 'https://slave.switzernet.com'; # Porta-Billing Admin Server
my $proxy_port = '8444';
my $uri_base = 'http://portaone.com/Porta/SOAP';
my $proxy = "$proxy_host:$proxy_port/soap/";
my %uris = (
  'Session' => "$uri_base/Session",
  'Account' => "$uri_base/Account",
  'Customer' => "$uri_base/Customer",
);
sub fault_handler {
  my ($soap, $res) = @_;
  die "SOAP Fault: $!, " . (ref $res ? $res->faultstring : $soap->transport->status);
}
my $session_service = SOAP::Lite
  ->uri($uris{'Session'})
  ->proxy($proxy)
  ->on_fault(\&fault_handler)
;
my $customer_service = SOAP::Lite
  ->uri($uris{'Customer'})
  ->proxy($proxy)
  ->on_fault(\&fault_handler)
;
my $account_service = SOAP::Lite
  ->uri($uris{'Account'})
  ->proxy($proxy)
  ->on_fault(\&fault_handler)
;

# required to support dateTime type
$session_service->serializer()
  ->xmlschema('http://www.w3.org/2001/XMLSchema');
$customer_service->serializer()
  ->xmlschema('http://www.w3.org/2001/XMLSchema');
$account_service->serializer()
```

Connection to Porta-Billing SOAP.

```
  ->xmlschema('http://www.w3.org/2001/XMLSchema');
my $LoginResponse = $session_service->login($soap_login, $soap_password);
my $session_id = $LoginResponse->result();
print "Logged in with session $session_id\n";
my $header = SOAP::Header->name('auth_info')->value({ session_id => $session_id });
```

```
my $max_credit_limit = 750;
my $req = '';


$req = 'SELECT
 c.i_customer,
 c.name,
 c.firstname,
 c.lastname,
 c.email,
 c.credit_limit,
 IF(2*c.credit_limit<2*i.previous_balance, if( 2*c.credit_limit>=' . $max_credit_limit
. ',' . $max_credit_limit . ', truncate(2*c.credit_limit,-1 )),
if(2*i.previous_balance >=' . $max_credit_limit . ',' . $max_credit_limit . ',
truncate(2*i.previous_balance,-1) ) ) as "newlimit",
 c.iso_4217,
 cn.notepad
FROM
 Customers c
INNER JOIN
 Invoices i
ON
  c.i_customer=i.i_customer
INNER JOIN
 Customer_Notepad cn
ON
  cn.i_customer=c.i_customer
WHERE
  c.i_rep=3
AND
```

The request for finding all customers who need to increase their credit limit.

`  c.credit_limit>50` `AND` `  i.previous_balance+i.payments<=0` `AND` `  100*(c.balance/c.credit_limit)>=70` `AND` `  c.credit_limit<2*i.previous_balance` `AND` `  i.issue_date>=(date_sub(now(), interval 1 month))` `HAVING` `  newlimit>c.credit_limit';`	
`my $customers = {};` `$customers = $dbh->selectall_hashref($req, 'i_customer');` `foreach my $i_customer (keys %$customers) {` `    my $line = '';` `    foreach my $keys (keys %{ $customers->{$i_customer} } ) {` `      $line = $line . $customers->{$i_customer}->{$keys}.",";` `    }` `    chop($line);` `    print LOGFILE '[', strftime("%Y-%m-%d %H:%M:%S", localtime) , '] ',  $line , "\n";` `}`	Printing all lines to the log file in order to backup data before we modify it.
`my $body_html = <<EMAIL_BODY_HTML;` `<html>` `  <head>` `    <style>` `    </style>` `  </head>` `  <body>` `<p>` `-- English version below` `</p>`	The email body.

```
<p>
Chère/Cher [[firstname]] [[lastname]],
</p>

<p>
En tenant compte de votre consommation téléphonique et de vos paiements du mois passé,
nous avons augmenté la limite de crédit de votre compte.<br>
Elle était à [[oldlimit]] [[currency_code]], et passe maintenant à [[newlimit]]
[[currency_code]].
</p>

<p>
Meilleures salutations,<br>
L’équipe Switzernet
</p>

<p>
----
</p>


<p>
Dear Customer,
</p>

<p>
After considering your phone line usage and your payments of the last month, we
increased the credit limit of your account.<br>
It was set to [[oldlimit]] [[currency_code]], and is now [[newlimit]]
[[currency_code]].
</p>

<p>
Best regards,<br>
The Switzernet team
</p>
  </body>
</html>
```

EMAIL_BODY_HTML	
`foreach my $i_customer (keys %$customers) {` `  my $customer_body = $body_html;` `  my $newlimit = $customers->{$i_customer}->{newlimit};` `  my $oldlimit = $customers->{$i_customer}->{credit_limit};` `  my $currency_code = $customers->{$i_customer}->{iso_4217};` `  my $firstname = $customers->{$i_customer}->{firstname};` `  my $lastname = $customers->{$i_customer}->{lastname};`	For each record found with the MySQL request we get the necessary values for the mail and notepad.
`  $newlimit = sprintf "%01.2f", $newlimit;` `  $oldlimit = sprintf "%01.2f", $oldlimit;` `  my $date = strftime "%y%m%d", localtime;` `  my $add_to_notepad = '<limit on='.$date.' from='.$oldlimit.' to='.$newlimit.' why=history />';`	The credit limits are formatted with to decimal. The data must match the notepad format. Creation of the new notepad line.
`  $customer_body =~ s/\[\[newlimit\]\]/$newlimit/g;` `  $customer_body =~ s/\[\[oldlimit\]\]/$oldlimit/g;` `  $customer_body =~ s/\[\[firstname\]\]/$firstname/g;` `  $customer_body =~ s/\[\[lastname\]\]/$lastname/g;` `  $customer_body =~ s/\[\[currency_code\]\]/$currency_code/g;`	Replacing the keywords in the customer body with the good values.

`` my $CustomerInfo = { 'i_customer' => $i_customer, 'credit_limit' => $customers->{$i_customer}->{newlimit} }; my $UpdateCustomerRequest = { 'customer_info' => $CustomerInfo }; my $AddUpdateCustomerResponse = $customer_service->update_customer($header , $UpdateCustomerRequest)->result; ```	Updating the customer limit with new value.
`` if (defined($AddUpdateCustomerResponse->{i_customer}) && $AddUpdateCustomerResponse->{i_customer} =~ /^[0-9]+$/ && $AddUpdateCustomerResponse->{i_customer} =~ /^[0-9]+$/ > 0){ my $sth = $dbh->prepare('UPDATE Customer_Notepad SET notepad = CONCAT(? , "\n" , notepad) WHERE i_customer = ?') or die "Couldn't prepare statement: " . $dbh->errstr; $sth->execute($add_to_notepad, $i_customer) or die "Couldn't execute statement: " . $sth->errstr; } ```	If the SOAP request return the i_customer, then we can update the database.
`` print "\nsending to ".$customers->{$i_customer}->{email}; my $msg = MIME::Lite->new(From => 'user@domain.com', To => $customers->{$i_customer}->{email}, Cc => 'user@domain.com', Subject => 'Augmentation de votre limite de crédit', Type => 'multipart/mixed'); $msg->attach(Type => 'text/html; charset=UTF-8', Data => $customer_body); ```	Sending the email info to internal and customer address.

`  $msg->send;` `}`	
`$dbh->disconnect();` `close (LOGFILE);` `print "\nEnd of the script\n";`	Closign database and log file.

Liens

Ce document :

http://switzernet.com/3/public/141015-automatic-web-subscriptions/

* * *