

AUTOMATIC ACCOUNT CREATION ON E-SIGN AND E-PAY

Document created on 2014-10-15

Nicolas Bondier

[pdf][doc][htm]

*

*

*



Copyright © 2014 by Switzernet

Contents

Introduction

Modifications on subscriptions web pages

 Email validation page

 Final subscription page (partial)

Script for managing web subscription and porta-gilling accounts

Adding porta-billing account creation to payment notifications pages

 PayPal notification page (partial)

 PostFinance notification page (partial)

Pages for manually enter payment and automatic account creation

 Main table page

 Page for opening prepaid account on manual payment

 Page for opening account on signed contract reception

Liens

Introduction

This document describes the scripts and web pages created to automate the creation of account when customer sign his contract with e-sign or by paying if the account is a prepaid account.

Modifications on subscriptions web pages

Email validation page

CODE	COMMENT
<pre><?php if(session_id() == '') {session_start();}</pre>	Session start
<pre>\$page_title = "Validating Subscribe"; ?> <html> <head> <?php include_once \$_SERVER['DOCUMENT_ROOT'].'/public/070608-subscribe/configure.php"; include_once \$_SERVER['DOCUMENT_ROOT'].'/public/070608- subscribe/includes/currency_prices.php"; include_once \$_SERVER['DOCUMENT_ROOT'].'/public/060801-web/includes/check_language.php"; include_once \$_SERVER['DOCUMENT_ROOT'].'/public/060801-web/includes/head2.php"; include_once \$_SERVER['DOCUMENT_ROOT'].'/public/060801-web/includes/dbconnect.php"; include_once \$_SERVER['DOCUMENT_ROOT'].'/public/060801-web/includes/functions.php"; //include_once "../060801-web/includes/tariff_graph.php"; include_once \$_SERVER['DOCUMENT_ROOT'].'/public/070608- subscribe/include/functions_communes.php"; include_once \$_SERVER['DOCUMENT_ROOT'].'/public/070608-subscribe/include/calcul_price.php";</pre>	Including all necessary files.
<pre>// DEFAULT LANGUAGE if (!isset(\$_SESSION['language'])) { if (constant("LANGUAGE_1") != '') { \$language = explode(';', constant("LANGUAGE_1")); \$_SESSION['language']['id'] = \$language[0]; \$_SESSION['language']['code'] = \$language[1]; } // If no language defined, "1;FR" is the default else { \$_SESSION['language']['id'] = '1'; \$_SESSION['language']['code'] = 'FR'; } }</pre>	Language file selector.

<pre> } } include_once \$_SERVER['DOCUMENT_ROOT']. "/public/060801-web/includes/languages/language_" . \$_SESSION['language']['code'] . ".php"; include_once \$_SERVER['DOCUMENT_ROOT']. "/public/070608- subscribe/includes/languages/language_" . \$_SESSION['language']['code'] . ".php"; </pre>	
<pre> \$mysqli = new mysqli(DB_SERVER,DB_SERVER_USERNAME,DB_SERVER_PASSWORD,DB_DATABASE); if (\$mysqli->connect_errno) { echo "Echec lors de la connexion à MySQL : (" . \$mysqli->connect_errno . ") " . \$mysqli- >connect_error; } </pre>	Opening MySQL connection.
<pre> if (isset(\$_POST['eMail'])){ //print_r(\$_POST); \$sql="INSERT INTO Web_Subscriptions (`Subscription_Timestamp` , `Account_Type` , `Device_Model` , `Currency` , `Title` , `Lastname` , `Firstname` , `Address` , `NPA` , `City` , `Country` , `Company_Name` , `Phone1` , `Phone2` , `Fax` , `Email` , `Referrer` , `Promotionnal_code` , `Comments` , `Email_Code` , `Paper_Invoice`) </pre>	The request for inserting the new subscription in MySQL.

<pre>) VALUES (NOW(),?)";</pre>	
<pre>\$Email_Code = freeRandomString(8);</pre>	Generating the Email validation code.
<pre>if (!isset(\$_POST['Type'])){ \$_POST['Type'] = NULL; } if (!isset(\$_POST['modele'])){ \$_POST['modele'] = NULL; } if (!isset(\$_SESSION['currency'])){ \$_SESSION['currency'] = NULL; } if (!isset(\$_POST['Titre'])){ \$_POST['Titre'] = NULL; } if (!isset(\$_POST['Nom'])){ \$_POST['Nom'] = NULL; } if (!isset(\$_POST['Prenom'])){ \$_POST['Prenom'] = NULL; } if (!isset(\$_POST['Adresse'])){ \$_POST['Adresse'] = NULL;</pre>	Verifying all filed before inserting in MySQL database.

```
}

if ( !isset( $_POST['NPA'] )){
    $_POST['NPA'] = NULL;
}

if ( !isset( $_POST['Ville'] )){
    $_POST['Ville'] = NULL;
}

if ( !isset( $_POST['Pays'] )){
    $_POST['Pays'] = NULL;
}

if ( !isset( $_POST['TelFixe'] )){
    $_POST['TelFixe'] = NULL;
}

if ( !isset( $_POST['TelMobile'] )){
    $_POST['TelMobile'] = NULL;
}

if ( !isset( $_POST['Fax'] )){
    $_POST['Fax'] = NULL;
}

if ( !isset( $_POST['eMail'] )){
    $_POST['eMail'] = NULL;
}

if ( !isset( $_POST['Parrain'] )){
    $_POST['Parrain'] = NULL;
}
```

```
if ( !isset( $_POST['promo_code'] )){
    $_POST['promo_code'] = NULL;
}

if ( !isset( $_POST['Comment'] )){
    $_POST['Comment'] = NULL;
}

if ( !isset( $_POST['societe'] )){
    $_POST['societe'] = NULL;
}

if ( !isset( $_POST['numero1'] )){
    $_POST['numero1'] = NULL;
}

if ( !isset( $_POST['numero2'] )){
    $_POST['numero2'] = NULL;
}

if ( !isset( $_POST['type_facture'] )){
    $_POST['type_facture'] = NULL;
}

preg_match_all('!\d+!', $_POST['TelFixe'], $matches);
$_POST['TelFixe'] = implode("", $matches[0]);

preg_match_all('!\d+!', $_POST['TelMobile'], $matches);
$_POST['TelMobile'] = implode("", $matches[0]);

preg_match_all('!\d+!', $_POST['Fax'], $matches);
```



```

$_POST['Fax'] = implode("", $matches[0]);

preg_match_all('!\d+!', $_POST['Parrain'], $matches);
$_POST['Parrain'] = implode("", $matches[0]);

$_POST['numero1'] = convert_to_41XXXXXXXXXX($_POST['numero1']);
$_POST['numero2'] = convert_to_41XXXXXXXXXX($_POST['numero2']);

$Paper_Invoice = 0;
if ( $_POST['type_facture'] == 'type_facture_poste' ){
    $Paper_Invoice = 1;
}

```

```

$stmt = $mysqli->prepare($sql);
if ( false=== $stmt ) {
    die('An error occured');
}

$stmt->bind_param('ssssssssssssssssssi',
    $_POST['Type'],
    $_POST['modele'],
    $_SESSION['currency'],
    $_POST['Titre'],
    $_POST['Nom'],
    $_POST['Prenom'],
    $_POST['Adresse'],
    $_POST['NPA'],
    $_POST['Ville'],
    $_POST['Pays'],
    $_POST['societe'],
    $_POST['TelFixe'],
    $_POST['TelMobile'],

```

Inserting
the
customer
and the
selected
numbers in
the MySQL
database.

```

$_POST['Fax'],
$_POST['eMail'],
$_POST['Parrain'],
$_POST['promo_code'],
$_POST['Comment'],
$Email_Code,
$Paper_Invoice
);

if ( false===$stmt ) {
    die('An error occurred');
}

$stmt->execute();

if ( false===$stmt ) {
    die('An error occurred');
}

$i_web_subscription = $mysqli->insert_id;

$stmt->close();

$sql = "INSERT INTO `Web_Subscription_Selected_Numbers` (
        h323_id,
        i_web_subscription,
        date
    ) VALUES (
        ?, ?, NOW()
    )";

for ($i=1; $i <= 2; $i++) {
    //echo $sql."<br>".$_POST['numero'.$i]."\n";

```

```

    if ( isset($_POST['numero'.$i]) && preg_match('/^[0-9]+$/',
$_POST['numero'.$i])){
        $stmt = $mysqli->prepare($sql);
        //echo $sql."<br>\n";
        if ( false=== $stmt ) {
            die('An error occured'. $mysqli->error);
        }
        $stmt->bind_param('ss',
            $_POST['numero'.$i],
            $i_web_subscription
        );
        if ( false=== $stmt ) {
            die('An error occured');
        }
        $stmt->execute();
        if ( false=== $stmt ) {
            die('An error occured');
        }
        $stmt->close();
    }
}

```

```

$email = $_POST['eMail'];
$emailSwitzernet = "contracts@switzernet.com";

$DEBUG = FALSE;

if ( $DEBUG ){
    $emailSwitzernet = 'user@domain.com';
}

```

Definition
of the email
recipients.

```

require('include/class.phpmailer.php');

$mail=new PHPMailer();
if ( ! isset( $_SESSION['language'] ) || ! isset( $_SESSION['language']['code'] ) ||
$_SESSION['language']['code'] != 'RU' ) {
    $mail->CharSet = 'ISO-8859-1';
} else {
    $mail->CharSet = 'UTF-8';
}

$mail->From=$emailSwitzernet;
$mail->FromName="Switzernet";
$mail->AddAddress($email);
$mail->AddBCC($emailSwitzernet);

//$mail->AddAddress($emailSwitzernet);
$mail->Subject = EMAIL_VALIDATION_EMAIL_CONFIRM;

$cher = "Cher(e)";
if ( $_POST['Titre'] == 'MONSIEUR' ){
    $cher = TRAITEMENT_WELCOME_BONJOUR;
} elseif ( $_POST['Titre'] == 'MADAME' ) {
    $cher = TRAITEMENT_WELCOME_BONJOURS;
}
//$Email_Code = strrev( implode('-', str_split(strrev($Email_Code), 3) ) );
$body = '
<center>
<table width="600px" cellspacing="0" cellpadding="20" style="border: none; background-
color: #FFFBFB; height:100%; padding:10px">
    <tr style="height:60px">
        <td style="text-align: left;vertical-align: top;">
            <a href="http://www.switzernet.com"><img style="width:150px"
src="" .base64_encode_image("images/switzernet.gif","gif").'" alt="switzernet"></img></a>
        </td>
    </tr>
    <tr style="height:40px">
        <td style="text-align: left;vertical-align: top;">

```

Sending
email to the
new
customer
with the
validation
code to
enter.

```

        <h3 style="font-weight: normal; color:black">' . EMAIL_VALIDATION_EMAIL_CONFIRM
    . '</h>
    </td>
</tr>
<tr style="height:60px">
    <td style="text-align: left;vertical-align: top;">
        <p>' . $cher . ' ' . $_POST['Prenom'] . ' ' . $_POST['Nom'] . ' ,</p>
        <p>' . EMAIL_VALIDATION_EMAIL_TEXT_1 . '</p>
    </td>
</tr>
<tr style="height:50px">
    <td style="text-align: center;vertical-align: top;">
        <a href="http://switzernet.com/public/070608-
subscribe/traitementInscription.php?em=' . urlencode($eMail) . '&cn=' . urlencode($Email_Code) . '">
        <span style="
        cursor: pointer;
        background-color: #0078e7;
        color: #fff;
        font-size: 100%;
        padding: .5em 1.5em;
        border: 0 rgba(0,0,0,0);
        text-decoration: none;
        border-radius: 2px;
        -webkit-transition: .1s linear -webkit-box-shadow;
        -moz-transition: .1s linear -moz-box-shadow;
        -ms-transition: .1s linear box-shadow;
        -o-transition: .1s linear box-shadow;
        transition: .1s linear box-shadow;
        display: inline-block;
        zoom: 1;
        line-height: normal;
        white-space: nowrap;
        vertical-align: baseline;
        text-align: center;
        -webkit-user-drag: none;
        -webkit-user-select: none;
        -moz-user-select: none;
        -ms-user-select: none;
        user-select: none;">
        ' . EMAIL_VALIDATION_EMAIL_BUTTON . '
    </span>
</a>
</td>

```

```

</tr>
<tr style="height:40px">
  <td style="text-align: center;vertical-align: top;">
    <p>' . EMAIL_VALIDATION_EMAIL_TEXT_2 .'</p>
  </td>
</tr>
<tr style="height:40px">
  <td style="text-align: left;vertical-align: top;">
    <p>' . EMAIL_VALIDATION_EMAIL_TEXT_3 .' <a style="color: #0078e7; text-decoration:
none;"
href="http://switzerland.com/email_validation">http://switzerland.com/email_validation</a> ' .
EMAIL_VALIDATION_EMAIL_TEXT_4 .' ' .strrev( implode('-', str_split(strrev($Email_Code), 3) )
).'</p>
  </td>
</tr>
<tr style="height:80px">
  <td style="text-align: left;vertical-align: top;">
    <p>' . EMAIL_VALIDATION_EMAIL_TEXT_5 .' <a style="color: #0078e7; text-decoration:
none;" href="http://switzerland.com" >www.switzerland.com</a>' . EMAIL_VALIDATION_EMAIL_TEXT_6
.'
    </p>
    <p>' . EMAIL_VALIDATION_EMAIL_BEST_REGARDS .'
    <br>' . EMAIL_VALIDATION_EMAIL_SWITZERNET_TEAM .'
    </p>
  </td>
</tr>
<tr>
  <td style="text-align: left;vertical-align: top;">
  </td>
</tr>
</table>
</center>

';

$mail->Body = $body;

$mail->IsHTML(true);
if(!$mail->Send())
{
  echo "Erreur !";
  exit;
}

```

}	
<pre>\$mysqli->close(); ?></pre>	Closing MySQL connection.
<pre><script language="javascript"> function traitement(cond,form) { if (cond.checked == 1){ my_form = eval(form) my_form.action="./validationEmail.php"; my_form.submit(); } else { alert("<?= VALIDATION_CG_ALERT ?>"); } } function retour(form) { my_form = eval(form) my_form.action="./index.php"; my_form.submit(); } </script> </head> <body> <center> <table style="width:850px;text-align:center;"> <tr> <td colspan="5"><? include "../060801-web/includes/top.php"; ?></td> </tr> <tr> <td style="text-align:left; ">
 <center> <h1><?= INSCR_TITLE ?></h1> </center></pre>	Form for entering the email code.

```

<h2><?= INSCR_PROG ?></h2>
  <?= INSCR_1 ?><?= $previous ?><?= INSCR_1bis ?>
  <br>
  <?= INSCR_2 ?>
  <br>
  <font color="#0078e7"><?= INSCR_3 ?></font>
  <br>
  <?= INSCR_4 ?>
</p>
<hr style="border: 0px;background-color: #D1D1D1;height: 2px; margin-top:10px;">
<br>
<center>
<h2><?= INSCR_3 ?></h2>
<br> <br>
<!--<span style=" font-size: 15px;">A confirmation email has been sent to '<?php
$_POST['eMail'] ?>'. Check your email to complete your subscription.</span-->
  <span style=" font-size: 15px; line-height:1.4";>
    <?php if ( isset($_POST['eMail']) && $_POST['eMail'] != '' && $_POST['eMail'] !=
NULL ){ ?>
      <?= EMAIL_VALIDATION_EMAIL_SENT_1 ?> '<?= $_POST['eMail'] ?>'.<br> <br>
    <?php } else { ?>
      <!--<?= EMAIL_VALIDATION_EMAIL_SENT_2 ?> <br>-->
    <?php } ?>
    <?= EMAIL_VALIDATION_EMAIL_VERIFY ?></span>
  <br>
  <br>
</p>
<form name="emailValidationForm" action="traitementInscription.php" method="POST">
  <table border="0" style="border-spacing: 5px;border-collapse: separate; font-
size: 15px;">
    <?php if ( !isset($_POST['eMail']) ){ ?>
    <tr>
      <td class="form_td" align="right">
        <?= EMAIL_VALIDATION_VALIDATION_EMAIL ?>
      </td><td><input class="input_form" type="text" name="eMail" maxlength="256"
value="" size="30"></td>
    </tr>
    <?php } ?>
    <tr>
      <td class="form_td" align="right">
        <?= EMAIL_VALIDATION_VALIDATION_CODE ?>
      </td>
    <td>

```



```

function freeRandomString($length = 5){
    global $mysqli;
    $used = array();

    $sql = "SELECT Email_Code FROM Web_Subscriptions WHERE active_Subscription = 1";
    $result = $mysqli->query($sql);
    while($row = $result->fetch_assoc()){
        $used[$row['Email_Code']] = TRUE;
    }
    if ($result === FALSE) {
        return FALSE;
    }
    $code = generateRandomString($length);
    while ( $used[$code] ) {
        $code = generateRandomString($length);
    }
    return $code;
}

```

Check if the random string exists.

```

function base64_encode_image ($filename,$filetype) {
    if ($filename) {
        $imgbinary = fread(fopen($filename, "r"), filesize($filename));
        return 'data:image/' . $filetype . ';base64,' . base64_encode($imgbinary);
    }
}

?>

```

Encode an image in base 64 for inserting the image in the email.

Final subscription page (partial)

CODE

COMMENT

<pre><?php if(session_id() == '') {session_start();}</pre>	Session start
<pre>\$debugMode=False; // Put False to check usual mode for debug user \$page_title = "Finished Subscribe";</pre>	Starting the session
<pre>\$ERROR_MESSAGE = ''; \$ERROR_TITLE = ''; \$ERROR = FALSE; ?></pre>	Preparing error variables.
<pre><html> <head> <!-- Piwik : Do not remove --> <script type="text/javascript"> var pkBaseURL = (("https:" == document.location.protocol) ? "https://switzernet.com/3/public/120427-site-stats/" : "http://switzernet.com/3/public/120427-site-stats/"); document.write(unescape("%3Cscript src='" + pkBaseURL + "piwik.js' type='text/javascript'%3E%3C/script%3E")); </script> <script type="text/javascript"> try { var piwikTracker = Piwik.getTracker(pkBaseURL + "piwik.php", 1); piwikTracker.setDocumentTitle("Subscribe end"); piwikTracker.trackPageView(); piwikTracker.enableLinkTracking(); } catch(err) {} </script> <noscript><p></p></noscript> <!-- End Piwik Tracking Code --></pre>	Javascript code for piwik.

```
<?php

$d = 0;

include_once $_SERVER['DOCUMENT_ROOT'].'/public/070608-subscribe/configure.php";

include_once $_SERVER['DOCUMENT_ROOT'].'/public/070608-
subscribe/includes/currency_prices.php";

include_once $_SERVER['DOCUMENT_ROOT'].'/public/070608-
subscribe/include/calcul_price.php";

include_once $_SERVER['DOCUMENT_ROOT'].'/public/060801-web/includes/dbconnect.php";

include_once $_SERVER['DOCUMENT_ROOT'].'/public/070608-
subscribe/include/functions_communes.php";

include_once $_SERVER['DOCUMENT_ROOT'].'/3/support/140805-post-address-validation-
management/include/configure.php";
```

Including
required PHP
script/pages.

[...]

Code cut here.

```
$subscription = array();
$modele       = "";
$type         = "";
$titre        = "";
$societe       = "";
$nom          = "";
$prenom       = "";
$adresse      = "";
$npa          = "";
$ville        = "";
$pays         = "";
$telFixe      = "";
$telMobile    = "";
$fax          = "";
```

Preparing the
variables of the
subscription.

<pre> \$eMail = ""; \$Parrain = ""; \$comment = ""; \$type_facture = ""; </pre>	
<pre> if ((!isset(\$_POST['emailValidationCode']) !isset(\$_POST['eMail'])) && (!isset(\$_GET['em']) !isset(\$_GET['cn']))){ \$ERROR = "WRONG_PAGE"; </pre>	Checking the POST and GET parameters. Creating a new error if these fields are not present.
<pre> } else { if (isset(\$_GET['em']) && isset(\$_GET['cn'])){ \$eMail = urldecode(\$_GET['em']); \$validationCode = urldecode(\$_GET['cn']); } elseif (isset(\$_POST['emailValidationCode']) && isset(\$_POST['eMail'])) { \$eMail = urldecode(\$_POST['eMail']); \$validationCode = urldecode(\$_POST['emailValidationCode']); } \$validationCode = preg_replace("/[^a-zA-Z0-9]+/", "", \$validationCode); //echo "eMail : ".\$eMail. "
"; //echo "validationCode : ".\$validationCode. "
"; \$sql = "SELECT i_web_subscription, verified_Email FROM Web_Subscriptions WHERE Email_Code = ? AND Email = ? LIMIT 1"; \$stmt = \$mysqli->prepare(\$sql); \$stmt->bind_param("ss", \$validationCode, \$eMail); \$stmt->execute(); \$stmt->bind_result(\$i_web_subscription, \$verified_Email); \$result = \$stmt->fetch(); \$stmt->close(); </pre>	Getting the email and email code from GET or POST. Keeping only alphanumeric fields of the code. Selecting in the database where Email_Code and Email are the same as the provided parameters.
<pre> if (\$result < 1){ \$ERROR = "WRONG_VALIDATION_CODE"; } else { if (\$verified_Email == 1){ </pre>	Checking if we have a result.

<pre> \$ERROR = "HAS_BEEN_VERIFIED_BEFORE"; } else { </pre>	
# [...] not commented	
<pre> function update_internal_subscription_mail_id(\$i_web_subscription,\$messageid){ if (! preg_match('/^[0-9]+\$/', \$i_web_subscription)) return FALSE; global \$mysqli; \$sql = "UPDATE Web_Subscriptions SET internal_subscription_mail_id = ? WHERE i_web_subscription = ?"; \$stmt = \$mysqli->prepare(\$sql); \$stmt->bind_param('si', \$messageid, \$i_web_subscription); \$stmt->execute(); \$stmt->close(); } </pre>	Function for updating the web subscription record with the message id used to send the email to internal mailbox.
<pre> function update_customer_subscription_mail_id(\$i_web_subscription, \$messageid){ if (! preg_match('/^[0-9]+\$/', \$i_web_subscription)) return FALSE; global \$mysqli; \$sql = "UPDATE Web_Subscriptions SET customer_subscription_mail_id = ? WHERE i_web_subscription = ?"; \$stmt = \$mysqli->prepare(\$sql); \$stmt->bind_param('si', \$messageid, \$i_web_subscription); \$stmt->execute(); \$stmt->close(); } </pre>	Function for updating the web subscription record with the message id used to send the email to customer mailboxes.
<pre> function update_esign_code(\$i_web_subscription,\$esign_code){ if (! preg_match('/^[0-9]+\$/', \$i_web_subscription)) return FALSE; global \$mysqli; \$sql = "UPDATE Web_Subscriptions SET Esign_Code = ? WHERE i_web_subscription = ?"; \$stmt = \$mysqli->prepare(\$sql); \$stmt->bind_param('si', \$esign_code, \$i_web_subscription); \$stmt->execute(); \$stmt->close(); } </pre>	Function for updating the web subscription record with a new esign code.

<pre> }</pre>	
<pre> function update_i_address_validation(\$i_web_subscription, \$i_address_validation){ if (! preg_match('/^[0-9]+\$/', \$i_web_subscription)) return FALSE; if (! preg_match('/^[0-9]+\$/', \$i_address_validation)) return FALSE; global \$mysqli; \$sql = "UPDATE Web_Subscriptions SET i_address_validation = ? WHERE i_web_subscription = ?"; \$stmt = \$mysqli->prepare(\$sql); \$stmt->bind_param('ii', \$i_address_validation, \$i_web_subscription); \$stmt->execute(); \$stmt->close(); } </pre>	Function for updating the web subscription with the id of the record inserted in the addresse validation table.
<pre> function update_web_login(\$i_web_subscription, \$web_login){ if (! preg_match('/^[0-9]+\$/', \$i_web_subscription)) return FALSE; global \$mysqli; \$sql = "UPDATE Web_Subscriptions SET web_login = ? WHERE i_web_subscription = ?"; \$stmt = \$mysqli->prepare(\$sql); \$stmt->bind_param('si', \$web_login, \$i_web_subscription); \$stmt->execute(); \$stmt->close(); } </pre>	Function to update the web login in subscription DB.
<pre> function update_web_password(\$i_web_subscription, \$web_password){ if (! preg_match('/^[0-9]+\$/', \$i_web_subscription)) return FALSE; global \$mysqli; \$sql = "UPDATE Web_Subscriptions SET web_password = ? WHERE i_web_subscription = ?"; \$stmt = \$mysqli->prepare(\$sql); \$stmt->bind_param('si', \$web_password, \$i_web_subscription); \$stmt->execute(); \$stmt->close(); } </pre>	Function to update the web password in subscription DB.
<pre> function update_h323_password(\$i_web_subscription_reserved_number, \$h323_password){ </pre>	Function to update the VoIP

```

    if ( ! preg_match('/^[0-9]+$/', $i_web_subscription_reserved_number) ) return FALSE;
    global $mysqli;
    $sql = "UPDATE Web_Subscription_Selected_Numbers SET h323_password = ? WHERE i_web_subscription_reserved_number = ?";
    $stmt = $mysqli->prepare($sql);
    $stmt->bind_param('si', $h323_password, $i_web_subscription_reserved_number);
    $stmt->execute();
    $stmt->close();
}

```

password
password in
subscription DB.

```

function update_reserved_numbers_i_web_subscription($number,$i_web_subscription){
    $number = convert_to_41XXXXXXXXXX($number);
    if ( ! preg_match('/^[0-9]+$/', $i_web_subscription) ) return FALSE;
    if ( ! preg_match('/^[0-9]+$/', $number) ) return FALSE;
    global $mysqli;
    $sql = "UPDATE reserved_numbers SET i_web_subscription = ? WHERE number = ?";
    $stmt = $mysqli->prepare($sql);
    $stmt->bind_param('is', $i_web_subscription, $number);
    $stmt->execute();
    $stmt->close();
}

```

Function to
update reserved
number in
subscription
table.

```

function update_subscription_language($i_web_subscription){
    if ( ! isset($_SESSION['language']['code']) ) return FALSE;
    global $mysqli;
    $sql = "UPDATE Web_Subscriptions SET subscription_language = ? WHERE i_web_subscription = ?";
    $stmt = $mysqli->prepare($sql);
    $stmt->bind_param('si', $_SESSION['language']['code'], $i_web_subscription);
    $stmt->execute();
    $stmt->close();
}

```

Function to
update language
in subscription
table.

```

function update_Total_order($i_web_subscription, $amount){
    global $mysqli;

```

Function to
update total
amount of the
order in


```

    $sql = "UPDATE Web_Subscriptions SET Total_order = ? WHERE i_web_subscription =
?";

    $stmt = $mysqli->prepare($sql);
    $stmt->bind_param('si', $amount, $i_web_subscription);
    $stmt->execute();
    $stmt->close();
}

```

subscription
table.

Script for managing web subscription and porta-gilling accounts

This script is a class that can be loaded from everywhere.

CODE	COMMENT
<pre> <?php class WebSubscriptions_PortaBilling_Management { </pre>	Getting information posted to this page.
<pre> private \$ServiceAccount; private \$ServiceCustomer; private \$LocalMysqli; private \$MasterMysqli; </pre>	Declaration to the private properties. These are the connections to MySQL and SOAP.
<pre> public \$web_subscription; public \$customer_info = array(); public \$account_list = array(); </pre>	Declaring the public properties that we will access later. \$customer_info is the data of a customer with a format near of the format required

	by porta-billing SOAP interface. \$account_list is a list of account info with a format near of the format required by porta-billing SOAP interface. \$web_subscription is the data from the website subscriptions.
<pre>function __construct(){ \$this->soap_connect(); \$this->web_db_connect(); \$this->master_db_connect(); }</pre>	We initiate the connections to MySQL and SOAP in the class constructor.
<pre>private function soap_connect(){ \$filename = "credentials/soap.conf.php"; if(!@include_once(\$filename)) throw new WebSubscriptions_PortaBilling_Management_Exception("Failed to include ". \$filename); \$filename = 'classes/PortaBillingSoapClient.php'; if(!@include_once(\$filename)) throw new WebSubscriptions_PortaBilling_Management_Exception("Failed to include ". \$filename); \$SOAPCredentials = new SOAPCredentials;</pre>	Connection to SOAP, Account and customer services.

```

$SOAP_user = $SOAPCredentials->get_user();
$SOAP_pass = $SOAPCredentials->get_password();

try {
    $this->ServiceCustomer = new
PortaBillingSoapClient('https://slave.switzernet.com:8444', 'Admin',
'Customer');
    $session_id = $this->ServiceCustomer->_login($SOAP_user, $SOAP_pass);
    $this->ServiceAccount = new
PortaBillingSoapClient('https://slave.switzernet.com:8444', 'Admin',
'Account');
    $session_id2 = $this->ServiceAccount->_login($SOAP_user, $SOAP_pass);
    $this->ServiceAccount->_setSessionId($session_id);
    return $this->session_id;
} catch (SoapFault $e) {
    throw new WebSubscriptions_PortaBilling_Management_Exception("Could not
connect to porta-billing SOAP server !");
}
}

```

```

private function web_db_connect(){
    $filename = 'connections/local.mysql.class.php';
    if(!@include_once( $filename ) ) throw new
WebSubscriptions_PortaBilling_Management_Exception( "Failed to include ".
$filename );

    try {
        $LocalMysql = new LocalMysql;
        $this->LocalMysql = $LocalMysql->connection;
        return $this->LocalMysql;
    } catch (mysqli_sql_exception $e) {
        throw new WebSubscriptions_PortaBilling_Management_Exception("Could not
connect to mysql server !". $e);
    }
}

```

Connection to the
local database.

<pre>}</pre>	
<pre>private function master_db_connect(){ \$filename = 'connections/master.mysql.class.php'; if(!@include_once(\$filename)) throw new WebSubscriptions_PortaBilling_Management_Exception("Failed to include ". \$filename); try { \$MasterMysql = new MasterMysql; \$this->MasterMysql = \$MasterMysql->connection; return \$this->MasterMysql; } catch (mysqli_sql_exception \$e) { throw new WebSubscriptions_PortaBilling_Management_Exception("Could not connect to mysql server !".\$e); } }</pre>	Connection to the master database.
<pre># Main public functions public function customer_and_account_must_be_created(){ \$this->get_web_subscription(); if (\$this->web_subscription['active_in_porta_billing'] != 0){ return FALSE; } if (\$this->web_subscription['verified_Email'] != 1){ return FALSE; } if (\$this->web_subscription['active_Subscription'] != 1){ return FALSE; } return TRUE; }</pre>	Function to check if we can create the customer according to the data we have. The subscription must not have been processed before, must be active and the email must be verified.
<pre>public function create_customer_and_account_from_web_subscription(){</pre>	The main function to call for creating

<pre> \$this->get_web_subscription(); if (\$this->web_subscription['active_in_porta_billing'] != 0){ throw new WebSubscriptions_PortaBilling_Management_Exception("Subscription " . \$this->web_subscription['i_web_subscription'] . " is present in porta-billing ! " . "\n\n" . \$this->print_r_this_true()); } if (\$this->web_subscription['verified_Email'] != 1){ throw new WebSubscriptions_PortaBilling_Management_Exception("Subscription " . \$this->web_subscription['i_web_subscription'] . " email's is not verified ! " . "\n\n" . \$this->print_r_this_true()); } if (\$this->web_subscription['active_Subscription'] != 1){ throw new WebSubscriptions_PortaBilling_Management_Exception("Subscription " . \$this->web_subscription['i_web_subscription'] . " subscription is no more active ! " . "\n\n" . \$this->print_r_this_true()); } \$this->create_customer_from_web_subscription(); \$this->create_accounts_from_web_subscription(); \$this->update_web_subscription_active_in_porta_billing(); \$this->deactivate_web_subscription_i_customer_active_Subscription(); return true; } </pre>	the customer and its accounts. It also update the local database when the subscription is done.
<pre> public function account_exists_in_pb(\$account_id){ \$GetAccountInfoRequest = array('id' => \$account_id); \$account_info = \$this->ServiceAccount->get_account_info(\$GetAccountInfoRequest); if (sizeof(\$account_info) > 0){ return TRUE; } return FALSE; } </pre>	Check if the account exists in the porta-billing.
<pre> public function customer_exists_in_pb(\$name = NULL, \$login = NULL){ if (\$name == NULL && \$login == NULL){ throw new WebSubscriptions_PortaBilling_Management_Exception("No name or login provided to check if customer exists !" . "\n\n" . \$this->print_r_this_true()); } if (\$name != NULL){ \$GetCustomerInfoRequest = array('name' => \$name); </pre>	Check if the customer exists in porta-billing. The name and login matters because there are unique.

```

        $customer_info = $this->ServiceCustomer-
>get_customer_info($GetCustomerInfoRequest);
        if ( sizeof($customer_info) > 0 ){ return TRUE; }
    }
    if ( $login != NULL ){
        $GetCustomerInfoRequest = array('login' => $login );
        $customer_info = $this->ServiceCustomer-
>get_customer_info($GetCustomerInfoRequest);
        if ( sizeof($customer_info) > 0 ){ return TRUE; }
    }
    return FALSE;
}

```

```

public function get_web_subscription(){

    $subscription = array();
    $sql = "SELECT * FROM Web_Subscriptions ws WHERE ws.i_web_subscription = ?";
    $stmt = $this->LocalMysqlLi->prepare($sql);
    $state = $stmt->bind_param( "i", $this->web_subscription['i_web_subscription']);
    $stmt->execute();
    $row=array();
    $this->stmt_bind_assoc($stmt, $row);
    while ($stmt->fetch()) {
        foreach ($row as $key => $value) {
            $subscription[$key] = $value;
        }
    }
    $stmt->close();
    $tmp = array();
    $sql = "SELECT * FROM Web_Subscription_Selected_Numbers WHERE i_web_subscription
= ?";
    $stmt = $this->LocalMysqlLi->prepare($sql);
    $stmt->bind_param( "i", $this->web_subscription['i_web_subscription']);
    $stmt->execute();
    $row=array();
    $this->stmt_bind_assoc($stmt, $row);
    while ($stmt->fetch()) {
        foreach ($row as $key => $value) {
            $tmp[$row['h323_id']] = $row;
        }
    }
}

```

Getting the web subscriptions from the local website and also the subscribed numbers.

<pre> \$stmt->close(); \$subscription['subscribedNumbers'] = array(); foreach (\$tmp as \$key => \$value) { array_push(\$subscription['subscribedNumbers'], \$value); } \$this->web_subscription = \$subscription; return \$subscription; } </pre>	
<pre> public function verify_esign_code(\$code){ \$this->get_web_subscription(); if (\$this->web_subscription['Esign_Code'] == \$code){ return true; } return false; } </pre>	Function to check if the esign code is valid.
<pre> public function set_i_web_subscription_by_account_number(\$number){ \$i_web_subscription = \$this->get_i_web_subscription_from_account_id(\$number); if (\$i_web_subscription){ \$this->set_i_web_subscription(\$i_web_subscription); return TRUE; } else { return FALSE; } } </pre>	Function to get the id of subscription from an account number.

```

    public function send_creation_notification( $validation_type,
$validation_data = '', $additionalData = '' ){
    # 1 : Connaitre le type de compte.
    # 2 : Si priv ou business envoyer.

    if(!@include( "/var/www/public/141005-
subscribe/includes/currency_prices.php" ) ) throw new
WebSubscriptions_PortaBilling_Management_Exception( "Failed to include
/var/www/public/141005-subscribe/includes/currency_prices.php" );

    $filename = 'locales/language_'. $this-
>web_subscription['subscription_language'].'.php';
    if (!file_exists($filename)) {
        $filename = '/var/www/public/141008-automatic-account-
creation/locales/language_FR.php';
    }
    if(!@include( $filename ) ) throw new
WebSubscriptions_PortaBilling_Management_Exception( "Failed to include ".
$filename );

    $filename = 'classes/PHPMailer/class.phpmailer.php';
    if(!@include_once( $filename ) ) throw new
WebSubscriptions_PortaBilling_Management_Exception( "Failed to include ".
$filename );

    $filename = 'classes/Parsedown.php';
    if(!@include_once( $filename ) ) throw new
WebSubscriptions_PortaBilling_Management_Exception( "Failed to include ".
$filename );

    $mail=new PHPMailer();
    $mail->CharSet = 'UTF-8';
    $mail->From = 'contracts@switzernet.com';

```

The function to send the email informing the account has been created. It replies back to the first email send from the website subscription pages.


```

$mail->AddAddress($this->web_subscription['Email']);
$mail->AddCustomHeader("References: <".$this->web_subscription['customer_subscription_mail_id']."@switzernet.com>");
$mail->AddCustomHeader("In-Reply-To: <".$this->web_subscription['customer_subscription_mail_id']."@switzernet.com>");
$mail->CharSet = 'UTF-8';

$XXX_XXX_XXX = '';
$XXXXXXXXXXXX = '';

//print_r($this);
foreach ( $this->web_subscription[ 'subscribedNumbers' ] as $account ) {
    //if ( $this->customer_exists_in_pb( NULL, $account['h323_id'] ) ){
        $XXX_XXX_XXX = $this->convert_to_0XX_XXX_XXX( $account['h323_id']
);
        $XXXXXXXXXXXX = $account['h323_id'];
        $voip_password = $account['h323_password'];
        break;
    //}
}

$serveur =
@file_get_contents("http://provisioning.switzernet.com/random_sip_server/");

if ( $serveur === FALSE ){
    $i = rand(1,40);
    $serveur = "sip".$i.".youroute.net";
}

$mail_Subject = '';
$mail_Body = '';

$timestamp = strtotime( $this->web_subscription['Subscription_Timestamp']
);

```

```

$subscription_date = date('Y-m-d',$timestamp);

$CREDIT_REFERRER = '';
$FIRST_CREDIT_LIMIT = '';

$CREDIT_REFERRER = $PRICES[$this->web_subscription['Currency']][ 'CREDIT_REFERRER' ];
$FIRST_CREDIT_LIMIT = $PRICES[$this->web_subscription['Currency']][ 'FIRST_CREDIT_LIMIT' ];

$body_variables = array(
    '[[language_code]]' => $this->web_subscription['subscription_language'],
    '[[0XX-XXX-XXXX]]' => $XXX_XXX_XXXX,
    '[[41XXXXXXXXXX]]' => $XXXXXXXXXXXX,
    '[[voip_password]]' => $voip_password,
    '[[web_login]]' => $this->web_subscription['web_login'],
    '[[web_password]]' => $this->web_subscription['web_password'],
    '[[voip_server]]' => $serveur,
    '[[Currency]]' => $this->web_subscription['Currency'],
    '[[FIRST_CREDIT_LIMIT]]' => $CREDIT_REFERRER,
    '[[CREDIT_REFERRER]]' => $FIRST_CREDIT_LIMIT
);

$mail_Subject = "[Signed] VOIP-" . $XXX_XXX_XXXX . " on " .
$subscription_date;

if ( $validation_type == "postal_validation" ){

    $mail_Subject = "[Signed postal] " . $XXX_XXX_XXXX . " on " .
$subscription_date;
    $mail_Body .= RECEIVED_POSTAL_VALIDATION;

} elseif ( $validation_type == "prepaid" ){

```

```
    $mail_Subject = "[Signed prepaid] VOIP-" . $XXX_XXX_XXXX . " on " .  
$subscription_date;  
    $mail_Body .= RECEIVED_PAYMENT_PREPAY;  
  
    } elseif ( $validation_type == "esign" ){  
  
        $mail_Subject = "[Signed esign] VOIP-" . $XXX_XXX_XXXX . " on " .  
$subscription_date;  
        $mail_Body .= RECEIVED_CONTRACT_ESIGN;  
  
    } elseif ( $validation_type == "contract" ){  
  
        $mail_Subject = "[Signed contract] VOIP-" . $XXX_XXX_XXXX . " on " .  
$subscription_date;  
        $mail_Body .= RECEIVED_CONTRACT_POST;  
  
    }  
  
    if ( $this->web_subscription['Device_Model'] == 'no' ){  
  
        $mail_Body .= ACCOUNT_OPEN_NO_DEVICE;  
  
    } else {  
  
        $mail_Body .= ACCOUNT_OPEN_WITH_DEVICE;  
  
    }  
  
    $mail_Body .= CREDENTIALS_VOIP;  
  
    $mail_Body .= LINK_DEVICES_CONFIGURATION;  
  
    $mail_Body .= CREDENTIALS_WEB_INTERFACE;
```

```
if ( $this->web_subscription['Account_Type'] != 'pre' ){
    $mail_Body .= INFO_CREDIT_LIMIT;
}

$mail_Body .= IMPORTANT_POINTS;

if ( $this->web_subscription['Account_Type'] != 'pre' ){
    $mail_Body .= IMPORTANT_POINT_INVOICES;
}

if ( $this->web_subscription['Account_Type'] != 'pre' && $this->web_subscription['Paper_Invoice'] == 1 ){
    $mail_Body .= IMPORTANT_POINT_INVOICES_PAPER;
}

$mail_Body .= IMPORTANT_POINT_HOW_TO_PAY;

if ( $this->web_subscription['Account_Type'] != 'pre' ){
    $mail_Body .= IMPORTANT_POINT_HOW_MUCH_TO_PAY_CREDIT;
} else {
    $mail_Body .= IMPORTANT_POINT_HOW_MUCH_TO_PAY_PREPAY;
}
```

```
$mail_Body .= IMPORTANT_POINT_SUPPORT;

$mail->Subject = $mail_Subject;

foreach ($body_variables as $occurence => $replacement) {
    $mail_Body = str_replace($occurence, $replacement, $mail_Body);
    # code...
}
$Parsedown = new Parsedown();

$html_mail_head = '<html>
<head>
  <title>' . $mail_Subject . '</title>
  <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
  <style>
    * {

    }
    body {
      background-color: white;
    }
    h1 {
      /*color: #da251d;*/
      font-size: 20px;
      font-weight: normal;
    }
    ul {
      padding-left: 5px;
      margin-left: 15px;
    }

    li { background-position: 50px 50px; list-style: none; }
```

```

    li li { list-style: disc; }
    a {
        color: #0078e7;
        text-decoration: none;
    }
    a:visited {

    }
</style>
</head>

<body>
<center>
<table width="600px" cellspacing="0" cellpadding="20" style="border: none;
background-color: white; height:100%; padding:10px">
    <tr style="height:60px">
        <td style="text-align: left;vertical-align: top;">
            <a href="http://www.switzernet.com"></img></a>
        </td>
    </tr>
    <tr style="height:60px">
        <td style="text-align: left;vertical-align: top;">'>
$html_mail_Body .= $Parsedown->text($mail_Body);
$html_mail_foot = SALUTATIONS.' </td>
    </tr>
</table>
</center>
</body>
</html>
';

$html_mail = $html_mail_head . $html_mail_Body . $html_mail_foot;

```

```

$mail->Body = $html_mail;
$mail->IsHTML(true);

$mail->Send();

# sending to contracts too with attachment.

$mail->ClearAllRecipients();
$mail->AddAddress('contracts@switzernet.com');
$attachment = $this->get_mail_notification_attachment_filename();
//echo "'".$attachment."'";
if ( $attachment != FALSE && file_exists( $attachment ) ){
    $mail->AddAttachment( $attachment );
}
$mail->send();
}

```

```

private function create_customer_from_web_subscription(){
    $this->get_web_subscription();
    $this->create_customer_info_from_web_subscription();
    $this->create_customer_in_pb();
    $this->fix_customer_info();
    $this->update_web_subscription_i_customer();
    $this->init_customer_notepad();
    $this->update_customer_notepad_with_account_creation();
    $this->charge_total_order();
    $this->create_customer_subscriptions_in_pb();
    $this->update_credit_limit_after_charges();
}

```

```

private function create_accounts_from_web_subscription(){
    $this->get_web_subscription();
}

```

Function that call
all needed
functions for
creating the
customer.

Function that call
all needed

<pre>\$this->create_accounts_list_from_web_subscription(); \$this->fix_accounts_info(); \$this->create_accounts_in_pb(); }</pre>	functions for creating the accounts.
<pre>private function update_web_subscription_active_in_porta_billing(){ \$sql = "UPDATE Web_Subscriptions SET active_in_porta_billing = 1 WHERE i_web_subscription = ?"; \$stmt = \$this->LocalMysqli->prepare(\$sql); \$state = \$stmt->bind_param("i", \$this->web_subscription['i_web_subscription']); \$stmt->execute(); \$stmt->close(); }</pre>	Function for updating the flag when the subscription has been added to porta-billing.
<pre>private function create_accounts_list_from_web_subscription(){ \$account_list = array(); \$filename = 'etc/products.conf.php'; if(!@include_once(\$filename)) throw new WebSubscriptions_PortaBilling_Management_Exception("Failed to include ". \$filename); foreach (\$this->web_subscription['subscribedNumbers'] as \$number) { \$account_info['id'] = \$number['h323_id']; \$account_info['billing_model'] = -1; \$account_info['i_customer'] = \$this->customer_info['i_customer']; \$account_info['iso_4217'] = \$this->web_subscription['Currency']; \$account_info['opening_balance'] = 0; \$account_info['balance'] = 0; \$account_info['activation_date'] = date('Y-m-d'); \$account_info['i_product'] = \$PRODUCT[\$this- >web_subscription['Currency']] [\$this->web_subscription['Account_Type']] ['i_product']; \$account_info['i_acl'] = 10007; \$account_info['i_lang'] = 'fr'; \$account_info['iso_639_1'] = 'fr'; \$account_info['service_flags'] = "N^N^N^NNN7^YN~N~NY^^NY~ N"; \$account_info['h323_password'] = \$number['h323_password']; \$account_info['follow_me_enabled'] = 'Y'; array_push(\$this->account_list , \$account_info); } }</pre>	This function retrieve the accounts order on the subscription web site and transform it in a list of account info that will be used by SOAP later.

<pre> private function fix_customer_info(){ \$sql = "UPDATE Customers SET service_flags = 'NNANNPN NNNNNN', i_acl = 10011 WHERE i_customer = ?"; \$stmt = \$this->MasterMysqli->prepare(\$sql); if (!\$stmt){ echo \$mysqli->error; } \$state = \$stmt->bind_param("i", \$this->customer_info['i_customer']); if (!\$state){ echo \$mysqli->error; } \$stmt->execute(); \$stmt->close(); } </pre>	After the customer has been created in the billing, the value of <code>service_flags</code> is incorrect. We update this with MySQL in the billing.
<pre> private function fix_accounts_info(){ \$sql = "UPDATE Accounts SET service_flags = 'N^^^NNN7^YN~N~NY^^NY~ N' WHERE i_customer = ?"; \$stmt = \$this->MasterMysqli->prepare(\$sql); \$state = \$stmt->bind_param("i", \$this->customer_info['i_customer']); \$stmt->execute(); \$stmt->close(); } </pre>	After the account has been created in the billing, the value of <code>service_flags</code> is incorrect. We update this with MySQL in the billing.
<pre> private function init_customer_notepad(){ \$sql = "INSERT INTO Customer_Notepad (i_customer) VALUES (?)"; \$stmt = \$this->MasterMysqli->prepare(\$sql); \$state = \$stmt->bind_param("i", \$this->customer_info['i_customer']); \$stmt->execute(); \$stmt->close(); } </pre>	This function create a customer notepad for the customer.
<pre> private function update_customer_notepad_with_account_creation(){ \$sql = "UPDATE Customer_Notepad SET notepad = '<order on=".date('ymd')."' />\n<post to=ch1015 on=".date('ymd')."' />' WHERE i_customer = ? LIMIT 1"; \$stmt = \$this->MasterMysqli->prepare(\$sql); } </pre>	This function write to the customer notepad the

```

$state = $stmt->bind_param( "i", $this->customer_info['i_customer'] );
$stmt->execute();
$stmt->close();
}

```

stamps we were manually inserting before.

```

private function create_customer_info_from_web_subscription(){
    if(!@include( "/var/www/public/141005-subscribe/includes/currency_prices.php" ) )
    throw new WebSubscriptions_PortaBilling_Management_Exception( "Failed to include
/var/www//public/141005-subscribe/includes/currency_prices.php" );
    # name
    if ( isset($this->web_subscription['Company_Name']) && strlen($this-
>web_subscription['Company_Name']) > 0 ){
        $this->customer_info['name'] = $this->web_subscription['Company_Name'];
    } else {
        $this->customer_info['name'] = $this->web_subscription['Firstname'] . " " .
$this->web_subscription['Lastname'];
    }
    if ( $this->customer_exists_in_pb( $this->customer_info['name'] ) ){
        $i = 1;
        while ( $this->customer_exists_in_pb( $this->customer_info['name'] . " " . $i ) ){
            $i++;
        }
        $this->customer_info['name'] = $this->customer_info['name'] . " " . $i;
    }
    $this->customer_info['i_customer_type']          = 1; # retail customer
    $this->customer_info['i_parent']                 = 0; # direct customer
    $this->customer_info['iso_4217']                 = $this-
>web_subscription['Currency'];
    $this->customer_info['opening_balance']          = 0;
    $this->customer_info['balance']                 = 0;
    $this->customer_info['i_billing_period']         = 4; # monthly
    $this->customer_info['i_acl']                    = 10011; # Switzernet Customer
    if ( $this->web_subscription['Currency'] == 'CHF' ){
        $this->customer_info['i_customer_class']     = 2;
    }
    if ( $this->web_subscription['Currency'] == 'EUR' ){
        $this->customer_info['i_customer_class']     = 5;
    }
    if ( $this->web_subscription['Currency'] == 'USD' ){
        $this->customer_info['i_customer_class']     = 6;
    }
}

```

This function transform the web subscription into customer info that we will use with SOAP for creating the customer.

```

$this->customer_info['i_acl'] = 3;
$this->customer_info['i_time_zone'] = 138;
$this->customer_info['service_flags'] = 'NNANNNPN NNNNNN';
$this->customer_info['companyname'] = $this-
>web_subscription['Company_Name'];
$salutations = array(
    'MADAME' => 'Mme.',
    'MONSIEUR' => 'M.',
    '' => '',
    NULL => ''
);
$this->customer_info['salutation'] = $salutations[ $this-
>web_subscription['Title'] ];
$this->customer_info['firstname'] = $this->web_subscription[
'Firstname' ];
$this->customer_info['lastname'] = $this->web_subscription[
'Lastname' ];
$words = split(' ', $this->web_subscription[ 'Address' ]);
$max = 41;
$lines = array();
$ln = 1;
foreach ( $words as $word ) {
    if ( !isset( $lines[$ln] ) ) { $lines[$ln] = '' ;}
    if ( strlen( $lines[$ln] . $word . " ") <= $max+1 ){
        $lines[$ln] .= $word . " ";
    } else {
        $ln++;
    }
}
for ($i=1; $i <= 5; $i++) {
    $this->customer_info['baddr'.$i] = rtrim($lines[$i], " ");
}
$this->customer_info['city'] = $this->web_subscription[ 'City'
];
$this->customer_info['zip'] = $this->web_subscription[ 'NPA'
];
$this->customer_info['country'] = $this->web_subscription[
'Country' ];
$this->customer_info['faxnum'] = $this->web_subscription[ 'Fax'
];
$this->customer_info['cont1'] = $this->web_subscription[
'Firstname' ] . " " . $this->web_subscription[ 'Lastname' ];

```

```

        $this->customer_info['phone1']                = $this->web_subscription[
'Phone1' ];
        $this->customer_info['phone2']                = $this->web_subscription[
'Phone2' ];
        $this->customer_info['email']                 = $this->web_subscription[
'Email' ];
        $this->customer_info['bcc']                   = 'billing@switzernet.com';
        $this->customer_info['i_lang']                 = strtolower( $this->
>web_subscription[ 'subscription_language' ] );
        $this->customer_info['login']                 = '';
        foreach ( $this->web_subscription[ 'subscribedNumbers' ] as $account ) {
            if ( ! $this->customer_exists_in_pb( NULL, $account['h323_id'] ) ){
                $this->customer_info['login'] = $account['h323_id'];
                break;
            }
        }
        if ( $this->customer_info['login'] == '' ){
            throw new WebSubscriptions_PortaBilling_Management_Exception( "No login can be
set for i_web_subscription ".$this->web_subscription['i_web_subscription'] . ". It
seems to be used!\n\n" . $this->print_r_this_true() );
        }
        $this->customer_info['password'] = $this->web_subscription['web_password'];
        if ( ( $this->web_subscription['Account_Type'] == 'priv' || $this->
>web_subscription['Account_Type'] == 'com' ) && isset( $PRICES[$this->
>web_subscription['Currency']][ 'FIRST_CREDIT_LIMIT' ] ) ){
            $this->customer_info['credit_limit']                =
$PRICES[$this->web_subscription['Currency']][ 'FIRST_CREDIT_LIMIT' ];
            $this->customer_info['credit_limit_warning']        = "80%";
        } else {
            $this->customer_info['credit_limit']                = 0;
            $this->customer_info['credit_limit_warning']        = -5;
        }
        $this->customer_info['ppm_enabled']                 = 'Y';
        $this->customer_info['callshop_enabled']            = 'N';
        $this->customer_info['drm_enabled']                 = 'N';
        $this->customer_info['cld_translation_rule']         =
's/^00//;s/^0/41/';
        $this->customer_info['cli_in_translation_rule']      = 's/^41/0/';
        return $this->customer_info;
    }

```

<pre> private function create_customer_in_pb(){ \$AddCustomerRequest = array('customer_info' => \$this->customer_info); try { \$AddUpdateCustomerResponse = \$this->ServiceCustomer->add_customer(\$AddCustomerRequest); \$this->customer_info['i_customer'] = \$AddUpdateCustomerResponse->i_customer; } catch (SoapFault \$e) { throw new WebSubscriptions_PortaBilling_Management_Exception("Unable to create customer !\n\n" . \$e . "\n\n" . \$this->print_r_this_true()); } return \$AddUpdateCustomerResponse; } </pre>	This function get the customer info we created and create the customer in the porta-billing with SOAP.
<pre> private function charge_total_order(){ if (\$this->web_subscription['Total_order'] > 0){ \$this->charge_customer(\$this->web_subscription['Total_order'], 'Total commande', 'subscription order'); } } </pre>	This function get the amount of the order and pass the amount to the function to charge the customer.
<pre> public function charge_customer(\$amount, \$visible_comment = 'Manual charge', \$internal_comment = 'Manual charge'){ \$MakeCustomerTransactionRequest = array('i_customer' => \$this->customer_info['i_customer'], 'action' => 'Manual charge', 'amount' => sprintf('%.5f', \$amount), 'visible_comment' => \$visible_comment, 'internal_comment' => \$internal_comment); try { \$MakeCustomerTransactionResponse = \$this->ServiceCustomer->make_transaction(\$MakeCustomerTransactionRequest); } catch (SoapFault \$e) { throw new WebSubscriptions_PortaBilling_Management_Exception("Unable to charge customer !\n\n" . \$e . "\n\n" . \$this->print_r_this_true()); } return \$MakeCustomerTransactionResponse; } </pre>	This public function charge the customer in porta-billing.

```

private function create_customer_subscriptions_in_pb(){
    $filename = 'etc/subscriptions.conf.php';
    if(!@include_once( $filename ) ) throw new
WebSubscriptions_PortaBilling_Management_Exception( "Failed to include ". $filename
);

    # line subscriptions
    $i_subscription = '';
    if ($this->web_subscription['Account_Type'] != 'pre' ) {
        $i_subscription = $SUBSCRIPTIONS[ $this->web_subscription['Currency'] ]
['normal_line'] ['i_subscription'];
    }
    $CustomerSubscriptionInfo = array(
        'i_subscription' => $i_subscription,
        'start_date' => date('Y-m-d'),
        'i_customer' => $this->customer_info['i_customer']
    );
    $AddCustomerSubscriptionRequest = array(
        'i_customer' => $this->customer_info['i_customer'],
        'subscription_info' => $CustomerSubscriptionInfo
    );
    try {
        $AddUpdateCustomerSubscriptionResponse = $this->ServiceCustomer-
>add_subscription( $AddCustomerSubscriptionRequest );
    } catch (SoapFault $e) {
        throw new WebSubscriptions_PortaBilling_Management_Exception("Unable to add
subscription line to customer !\n\n" . $e . "\n\n" . $this->print_r_this_true() );
    }

    # line subscriptions

    $i_subscription = '';
    if ($this->web_subscription['Paper_Invoice'] == 1 ) {

        $i_subscription = $SUBSCRIPTIONS[ $this->web_subscription['Currency'] ]
['paper_invoice'] ['i_subscription'];

        $CustomerSubscriptionInfo = array(
            'i_subscription' => $i_subscription,
            'start_date' => date('Y-m-d'),
            'i_customer' => $this->customer_info['i_customer']
        );
    }
}

```

This function manage to add the correct subscriptions customer to porta-billing.

There are the subscription for the lines and the subscription for the paper invoice if it is set.

```

        $AddCustomerSubscriptionRequest = array(
            'i_customer' => $this->customer_info['i_customer'],
            'subscription_info' => $CustomerSubscriptionInfo
        );

        try {
            $AddUpdateCustomerSubscriptionResponse = $this->ServiceCustomer-
>add_subscription( $AddCustomerSubscriptionRequest );
        } catch (SoapFault $e) {
            throw new WebSubscriptions_PortaBilling_Management_Exception("Unable to add
subscription line to customer !\n\n" . $e . "\n\n" . $this->print_r_this_true() );
        }

    }

    return TRUE;
}

```

```

private function update_credit_limit_after_charges(){
    if(!@include( "/var/www/public/141005-subscribe/includes/currency_prices.php" ) )
    throw new WebSubscriptions_PortaBilling_Management_Exception( "Failed to include
/var/www/public/141005-subscribe/includes/currency_prices.php" );

    if ( ( $this->web_subscription['Account_Type'] == 'priv' || $this-
>web_subscription['Account_Type'] == 'com' ) && isset( $PRICES[$this-
>web_subscription['Currency']][ 'FIRST_CREDIT_LIMIT' ] ) ){
        $this->customer_info['credit_limit'] = $this->web_subscription['Total_order'] +
$PRICES[$this->web_subscription['Currency']][ 'FIRST_CREDIT_LIMIT' ];
        $this->update_credit_limit();
    }
}

```

Usually, when we charge an account (after a device order), if this account is business or private, we have to update the credit limit to permit to user to call.

Here we update the credit limit in the customer info with the order amount order.

<pre> private function update_credit_limit(){ \$CustomerInfo = array('i_customer' => \$this->customer_info['i_customer'], 'credit_limit' => sprintf('%.5f', \$this->customer_info['credit_limit'])); \$UpdateCustomerRequest = array('customer_info' => \$CustomerInfo); try { \$AddUpdateCustomerResponse = \$this->ServiceCustomer->update_customer(\$UpdateCustomerRequest); } catch (SoapFault \$e) { throw new WebSubscriptions_PortaBilling_Management_Exception("Unable to update credit limit to customer !\n\n" . \$e . "\n\n" . \$this->print_r_this_true()); } return TRUE; } </pre>	This apply the update of the credit limit to the customer in porta-billing.
<pre> private function update_web_subscription_i_customer(){ \$sql = "UPDATE Web_Subscriptions SET pb_i_customer = ? WHERE i_web_subscription = ?"; \$stmt = \$this->LocalMysqli->prepare(\$sql); \$state = \$stmt->bind_param("ii", \$this->customer_info['i_customer'], \$this- >web_subscription['i_web_subscription']); \$stmt->execute(); \$stmt->close(); } </pre>	This function updates the local MySQL database with the new <code>i_customer</code> of the customer inserted in the porta-billing.
<pre> private function deactivate_web_subscription_i_customer_active_Subscription(){ \$sql = "UPDATE Web_Subscriptions SET active_Subscription = 0 WHERE i_web_subscription = ?"; \$stmt = \$this->LocalMysqli->prepare(\$sql); \$state = \$stmt->bind_param("i", \$this->web_subscription['i_web_subscription']); \$stmt->execute(); \$stmt->close(); } </pre>	Function for disabling a valid web subscription.
<pre> private function create_accounts_in_pb(){ \$AddAccountResponses = array(); foreach (\$this->account_list as \$account_info) { \$AddAccountRequest = array('account_info' => \$account_info); try { </pre>	This function create the account in porta-billing with SOAP.

<pre> \$AddAccountResponse = \$this->ServiceAccount->add_account(\$AddAccountRequest); array_push(\$AddAccountResponses, \$AddAccountResponse); } catch (SoapFault \$e) { throw new WebSubscriptions_PortaBilling_Management_Exception("Unable to create account !\n\n" . \$e . "\n\n" . \$this->print_r_this_true()); } } return \$AddAccountResponse; } </pre>	
<pre> private function stmt_bind_assoc (&\$stmt, &\$out) { global \$mysqli; \$data = \$stmt->result_metadata(); \$fields = array(); \$out = array(); \$fields[0] = \$stmt; \$count = 1; while(\$field = \$data->fetch_field()) { \$fields[\$count] = &\$out[\$field->name]; \$count++; } call_user_func_array(mysqli_stmt_bind_result, \$fields); } </pre>	Function to build an array with MySQL results.
<pre> private function get_i_web_subscription_from_account_id(\$number){ \$sql = "SELECT i_web_subscription FROM reserved_numbers WHERE number = ?"; \$stmt = \$this->LocalMysqli->prepare(\$sql); \$state = \$stmt->bind_param("s", \$number); \$stmt->execute(); \$this->stmt_bind_assoc(\$stmt, \$res); \$stmt->fetch(); \$stmt->close(); if (\$res['i_web_subscription'] != 0){ return \$res['i_web_subscription']; } return false; } </pre>	Getting a web subscription in the local database from an ordered account id.

<pre>private function print_r_this_true(){ return print_r(\$this, TRUE); }</pre>	Function for printing all class data. Used for debugging.
<pre>private function convert_to_0XX_XXX_XXXX(\$number){ \$number = ereg_replace("[^0-9]", "", \$number); if (ereg('^41[0-9]{9}\$', \$number)){ \$number=substr_replace(substr_replace(ereg_replace("^41", "0", \$number), "-", 3, 0), "-", 7, 0); return \$number; } elseif (ereg('^0[0-9]{9}\$', \$number)) { \$number=substr_replace(substr_replace(\$number, "-", 3, 0), "-", 7, 0); return \$number; } else { return FALSE; } }</pre>	Function for converting any number string to the following format 0XX XXX XXX.
<pre>private function convert_to_41XXXXXXXXXX(\$number){ \$number = ereg_replace("[^0-9]", "", \$number); if (ereg('^0[0-9]{9}\$', \$number)){ \$number=ereg_replace("^0", "41", \$number); return \$number; } elseif (ereg('^41[0-9]{9}\$', \$number)) { return \$number; } else { return FALSE; } }</pre>	Function for converting any number string to the following format 41XX XXX XXX.
<pre># Setters and getters public function get_all_variables(){ return \$this; }</pre>	Function for returning the whole class.

<pre>public function set_i_web_subscription(\$i_web_subscription){ \$this->web_subscription['i_web_subscription'] = \$i_web_subscription; }</pre>	Function for selecting the web subscription.
<pre>private function set_web_subscription(\$web_subscription){ \$this->web_subscription = \$web_subscription; }</pre>	Function for loading the web subscription data.
<pre>private function set_customer_info(\$customer_info){ \$this->customer_info = \$customer_info; }</pre>	Function for setting the customer info (format) to the current class.
<pre>private function set_account_list(\$account_list){ \$this->account_list = \$account_list; }</pre>	Function for setting the account list to the current class.
<pre># Closing functions public function _close(){ \$this->LocalMysql->close(); \$this->MasterMysql->close(); \$this->__destruct(); }</pre>	Function to call to close the the class.
<pre>function __destruct(){ \$this->ServiceCustomer->_logout(); \$this->ServiceAccount->_logout(); }</pre>	Destructor, which is called everytime when the class instance is closed.
<pre>}</pre>	End of the class.
<pre>class WebSubscriptions_PortaBilling_Management_Exception extends Exception{</pre>	Extended exception class

[1'e9q'1 auto signup]

```
public function __construct($message = null, $code = 0, Exception $previous = null)
{
    try {
        mail('user@domain.com',
            'Automatic account management error',
            $message . "\n\n" . "Message sent from web server :" . __FILE__);
    } catch(Exception $e) {
        echo $e."\n";
    }
}
?>
```

that also send an email when a problem occurs.

Adding porta-billing account creation to payment notifications pages

PayPal notification page (partial)

CODE

```
include_once
"classes/WebSubscriptions_PortaBilling_Management/WebSubscriptions_PortaBilling_Management.php";

[...]
```

```
if ( $payment_status == 'Completed' ){
```

COMMENT

Session start

This is when the payment is ok. Account creation

```

$wpbm = new WebSubscriptions_PortaBilling_Management();
if ( isset($_POST['orderID']) && $_POST['orderID'] != '' ){

    $Esign = '';
    if ( preg_match('/^VOIP-([^-]+-[^-]+-[^-]+)-([^-]+)$/ ', $_POST['orderID'], $matches,
PREG_OFFSET_CAPTURE) ){
        $item_number = $matches[1][0];
        $Esign = $matches[2][0];
    }

    if ( preg_match('/^VOIP-([^-]+-[^-]+-[^-]+)$/ ', $_POST['orderID'], $matches, PREG_OFFSET_CAPTURE) ){
        $item_number = $matches[1][0];
    }

    $item_number = format_account($item_number);

    $r = $wpbm->set_i_web_subscription_by_account_number( $item_number );
    if ( $r ){
        $creation = false;
        if ( ! $wpbm->account_exists_in_pb( $item_number ) && $wpbm->customer_and_account_must_be_created()
){
            $wpbm->get_web_subscription( );
            if ( $wpbm->web_subscription['Account_Type'] == 'pre' ){
                $creation = $wpbm->create_customer_and_account_from_web_subscription();
                if ( $creation == TRUE ){
                    $wpbm->send_creation_notification("prepaid");
                }
            } elseif ( $Esign != '' && $wpbm->verify_esign_code( $Esign ) ){
                $creation = $wpbm->create_customer_and_account_from_web_subscription();
                if ( $creation == TRUE ){
                    $wpbm->send_creation_notification("esign");
                }
            } else {
                mail($mail_To, "WRONG ESIGN RECEIVED", print_r( $wpbm->get_all_variables() ) );
            }
        }
    }
}
}
}

```

comes
after.

Getting
the order.
We get
the esign-
code if
present.

If the
account is
prepaid,
we create
the
account.
Else, we
get the
Esign
code and
verify if it
is correct.

PostFinance notification page (partial)

CODE	COMMENT
<pre>include_once "classes/WebSubscriptions_PortaBilling_Management/WebSubscriptions_PortaBilling_Management.php";</pre>	Session start
<pre>[...] if (\$_POST['STATUS'] == '9'){</pre>	This is when the payment is ok. Account creation comes after.
<pre> \$wpbm = new WebSubscriptions_PortaBilling_Management(); if (isset(\$_POST['orderId']) && \$_POST['orderId'] != ''){ \$Esign = ''; \$item_number = ''; if (preg_match('/^VOIP-([^-]+-[^-]+-[^-]+)-([^-]+)\$/', \$_POST['orderId'], \$matches, PREG_OFFSET_CAPTURE)){ \$item_number = \$matches[1][0]; \$Esign = \$matches[2][0]; } if (preg_match('/^VOIP-([^-]+-[^-]+-[^-]+)\$/', \$_POST['orderId'], \$matches, PREG_OFFSET_CAPTURE)){ \$item_number = \$matches[1][0]; } }</pre>	Getting the order. We get the esign-code if present. If the account is prepaid, we create the account. Else, we get the Esign code and

```

$item_number = format_account($item_number);

$r = $wpbm->set_i_web_subscription_by_account_number( $item_number );
if ( $r ){
    $creation = false;
    if ( ! $wpbm->account_exists_in_pb( $item_number ) && $wpbm-
>customer_and_account_must_be_created() ){
        $wpbm->get_web_subscription( );
        if ( $wpbm->web_subscription['Account_Type'] == 'pre' ){
            $creation = $wpbm->create_customer_and_account_from_web_subscription();
            if ( $creation == TRUE ){
                $wpbm->send_creation_notification("prepaid");
            }
        } elseif ( $Esign != '' && $wpbm->verify_esign_code( $Esign ) ){
            $creation = $wpbm->create_customer_and_account_from_web_subscription();
            if ( $creation == TRUE ){
                $wpbm->send_creation_notification("esign");
            }
        } else {
            mail($mail_To, "WRONG ESIGN RECEIVED", print_r( $wpbm->get_all_variables() ) );
        }
    }
}
}
}

```

verify if it
is correct.

Pages for manually enter payment and automatic account creation

Main table page

CODE	COMMENT
<pre><?php if(session_id() == '') {session_start();} # defined default here. \$_SESSION['WS_TABLE_PARAMS'] = array(); \$_SESSION['WS_TABLE_PARAMS']['display_columns'] = array('Accounts' , 'Account_Type' , 'Subscription_Timestamp' , 'Company' , 'Firstname' , 'Lastname' , 'Address' , 'NPA' , 'City' , 'Country' , 'Email' , 'Referrer' , 'Promotionnal_code' , 'Comments'); \$_SESSION['WS_TABLE_PARAMS']['account_type_display'] = array('pre' , 'priv' , 'com'); \$_SESSION['WS_TABLE_PARAMS']['subscription_status_display'] = array('pending');</pre>	Session start Definition of default values in session if the session does not exists.

<pre> \$_SESSION['WS_TABLE_PARAMS']['page'] = 1; \$_SESSION['WS_TABLE_PARAMS']['results_per_page'] = 10; \$_SESSION['WS_TABLE_PARAMS']['order_by'] = 'Subscription_Timestamp'; \$_SESSION['WS_TABLE_PARAMS']['order_direction'] = 'DESC'; </pre>	
<pre> if (isset(\$_POST['action']) && \$_POST['action'] == 'new_pref'){ \$_SESSION['WS_TABLE_PARAMS'] = \$_POST; header("Location: " . "http://" . \$_SERVER['HTTP_HOST'] . dirname(\$_SERVER['SCRIPT_NAME'])); } if (is_array(\$_GET) && count(\$_GET) > 0){ \$_SESSION['WS_TABLE_PARAMS'] = array_merge(\$_SESSION['WS_TABLE_PARAMS'], \$_GET); header("Location: " . "http://" . \$_SERVER['HTTP_HOST'] . dirname(\$_SERVER['SCRIPT_NAME'])); } </pre>	Getting all post and get parameters. Then saving to the session and reloading the page (this way we do not have anymore reloading alert if we reload and press back button in browser).
<pre> require_once 'connections/local.mysql.class.php'; \$LocalMysqli = new LocalMysqli; \$mysqli = \$LocalMysqli->connection; # Availables parameters to pass in the SQL command : \$available_Columns = array(); \$sql= "SELECT `COLUMN_NAME` FROM `INFORMATION_SCHEMA`.`COLUMNS` WHERE `TABLE_SCHEMA`='switzern_smallbiz' AND `TABLE_NAME`='Web_Subscriptions'"; if (\$result = \$mysqli->query(\$sql)) { while (\$row = \$result->fetch_assoc()) { </pre>	Selecting column names of the table that will be displayed.

<pre> \$available_Columns[\$row['COLUMN_NAME']] = 1; } } \$available_Columns['Accounts'] = 1; </pre>	
<pre> unset(\$available_Columns['web_password']); unset(\$available_Columns['internal_subscription_mail_id']); unset(\$available_Columns['customer_subscription_mail_id']); unset(\$available_Columns['activation_file']); unset(\$available_Columns['web_password']); </pre>	Removing sensible data from the columns to display.
<pre> \$available_Account_Types = array('pre' => 'Prepaid', 'priv' => 'Private', 'com' => 'Business'); \$available_Subscription_Statuses = array('pending' => 'Pending', 'processed' => 'Processed', 'deleted' => 'Deleted'); </pre>	Definition of the available accounts types and subscription statuses.
<pre> # Intitiate variables \$display_columns = array(); \$page_to_display = 1; \$results_per_page = 10; \$order_by = 'Subscription_Timestamp'; \$order_direction = 'DESC'; \$account_type_display = array(); \$subscription_status_display = array(); </pre>	Initialization of the parameters that will be used for displaying table.

```

if ( isset( $_SESSION['WS_TABLE_PARAMS']['action'] ) &&
$_SESSION['WS_TABLE_PARAMS']['action'] == 'new_pref' ){

    if ( isset( $_SESSION['WS_TABLE_PARAMS']['display_columns'] ) ){
        $display_columns = $_SESSION['WS_TABLE_PARAMS']['display_columns'];
    }

    if ( isset( $_SESSION['WS_TABLE_PARAMS']['page'] ) ){
        $page_to_display = $_SESSION['WS_TABLE_PARAMS']['page'];
    }

    if ( isset( $_SESSION['WS_TABLE_PARAMS']['order_direction'] ) ){
        if ( $_SESSION['WS_TABLE_PARAMS']['order_direction'] == 'ba' ){
            $order_direction = 'DESC';
        } else {
            $order_direction = '';
        }
    }

    if ( isset( $_SESSION['WS_TABLE_PARAMS']['results_per_page'] ) && preg_match('/^[0-9]+$/', $_SESSION['WS_TABLE_PARAMS']['results_per_page']) ){
        $results_per_page = $_SESSION['WS_TABLE_PARAMS']['results_per_page'];
    }

    if ( isset( $_SESSION['WS_TABLE_PARAMS']['order_by'] ) ){
        if ( isset( $available_Columns[ $_SESSION['WS_TABLE_PARAMS']['order_by'] ] ) ){
            $order_by = $_SESSION['WS_TABLE_PARAMS']['order_by'];
        }
    }

    if ( isset( $_SESSION['WS_TABLE_PARAMS']['account_type_display'] ) ){
        foreach ( $_SESSION['WS_TABLE_PARAMS']['account_type_display'] as $value ) {
            if ( isset($available_Account_Types[$value]) ){
                $account_type_display[$value] = 1;
            }
        }
    }

    if ( isset( $_SESSION['WS_TABLE_PARAMS']['subscription_status_display'] ) ){

```

If action is to save new displaying preferences, we read the session variables.

<pre> foreach (\$_SESSION['WS_TABLE_PARAMS']['subscription_status_display'] as \$value) { if (isset(\$available_Subscription_Statuses[\$value])){ \$subscription_status_display[\$value] = 1; } } } } </pre>	
<pre> \$columns_to_display = array(); foreach (\$display_columns as \$name) { if (isset(\$available_Columns[\$name])){ array_push(\$columns_to_display, \$name); } } </pre>	Removing columns that are not available.
<pre> # order by and order direction \$order_string = " ORDER BY wb.\$order_by \$order_direction "; </pre>	The order string construction.
<pre> # limit results per page \$offset = (\$page_to_display - 1) * \$results_per_page; \$limit_offset_string = " LIMIT \$results_per_page OFFSET \$offset"; </pre>	We limit results per page with the user settings and we set the offset from which record we have to display result, which depends of the current page.
<pre> # filter account type \$conditions = array(); </pre>	Creating an array the will contain all condition string for building

	the MySQL request.
<pre>foreach (\$available_Account_Types as \$key => \$value) { if (!isset(\$account_type_display[\$key])){ array_push(\$conditions, "Account_Type != '". \$key. "'"); } }</pre>	Setting the accounts type to display. There are prepaid, private and business. We mention to not show the account type not choosen because byy default MySQL will show all.
<pre>array_push(\$conditions, "verified_Email = 1");</pre>	Email code must be verified.
<pre>\$subscription_status_display_string = ''; \$str = ''; if (isset(\$subscription_status_display['pending'])){ \$str .= 'OR (active_Subscription = 1 AND active_in_porta_billing = 0) '; } if (isset(\$subscription_status_display['processed'])){ \$str .= 'OR (active_Subscription = 0 AND active_in_porta_billing = 1) '; } if (isset(\$subscription_status_display['deleted'])){ \$str .= 'OR (active_Subscription = 0 AND active_in_porta_billing = 0) '; } if (\$str != ''){ if (preg_match('/^OR(.+)/', \$str, \$matches)){ \$subscription_status_display_string = \$matches[1]; array_push(\$conditions, "(".\$subscription_status_display_string.")"); } }</pre>	Condition for retieving pending, precessed and deleted subscriptions.

```

$i = 0;
$conditions_string = '';
foreach ($conditions as $value) {
    if ( $value != '' ){
        if ( $i > 0 ){ $conditions_string = $conditions_string . ' AND ' ;}
        $conditions_string = $conditions_string . $value;
        $i++;
    }
}

```

We concatenate all conditions together separated by an AND keyword.

```

$total_web_subscriptions = 0;
$sql = "SELECT wb.i_web_subscription
        FROM Web_Subscriptions wb
        INNER JOIN Web_Subscription_Selected_Numbers wssn
        ON wb.i_web_subscription = wssn.i_web_subscription
        WHERE $conditions_string GROUP BY wb.i_web_subscription";
if ($result = $mysqli->query($sql)) {
    $total_web_subscriptions = $result->num_rows;
}
$min_pages = 1;
$max_pages = ceil($total_web_subscriptions / $results_per_page);

```

Just counting the total number of results to get the number of pages to display.

```

$sql = "SELECT *, GROUP_CONCAT(wssn.h323_id) AS Accounts
        FROM Web_Subscriptions wb
        INNER JOIN Web_Subscription_Selected_Numbers wssn
        ON wb.i_web_subscription = wssn.i_web_subscription
        WHERE $conditions_string GROUP BY wb.i_web_subscription $order_string
        $limit_offset_string";

$web_subscriptions = array();

$account_type_converter = array('com' => 'business',
                                'priv' => 'private',
                                'pre' => 'prepaid');

```

Selecting the matching subscriptions.

```

if ($result = $mysqli->query($sql)) {
    while ($row = $result->fetch_assoc()) {
        if ( isset( $row['Account_Type'] ) && isset($account_type_converter[
$row['Account_Type'] ])) ){
            $row['Account_Type'] = $account_type_converter[ $row['Account_Type'] ];
        }
        array_push($web_subscriptions, $row);
    }
}

```

```

//echo $sql;
$ROOT_DIR = dirname($_SERVER['SCRIPT_NAME']);

?>

```

Definition of the root directory. We need it to retrieve documents on web server.

```

<!doctype html>
<html>
<head>
    <meta http-equiv="X-UA-Compatible" content="IE=Edge;chrome=1" />
    <meta http-equiv="Content-Type" content="text/html; charset=utf-8">
    <title>Web subscriptions management</title>
    <link rel="stylesheet" href="css/pure-min.css">
    <link rel="stylesheet" href="css/styles.css">
    <script src="js/jquery-1.11.1.min.js"></script>
    <!--[if lt IE 7]>
    <script src="http://ie7-js.googlecode.com/svn/version/2.1(beta4)/IE7.js"></script>
    <![endif]-->
</head>
<body>
    <div id="head" style="padding:15px; width:80%; margin: 0 auto;" >
        <!--  -->
        <h1 style="text-align:center;margin-top:20px;">Web subscriptions management</h1>
    </div>
    <div id="container">
    <div id="table-wrapper">

```

HTML heading.

<pre> <table id="table" class="pure-table"> <thead> <tr> <?php foreach (\$columns_to_display as \$value) { ?? <th name="<?=\$value ?>" title="<?=\$value ?>"><?=\$value ?></th> <?php } ?? <th name="" title="">Actions</th> </tr> </thead> </pre>	Table header. We display the selected column fields.
<pre> <tbody> <?php foreach (\$web_subscriptions as \$web_subscription) { ?? <tr> <?php foreach (\$columns_to_display as \$column_name) { ?? <td><?=\$web_subscription[\$column_name] ?></td> <?php } ?? </pre>	Then for each subscription, we display the values.
<pre> <td> <?php if (\$web_subscription['Account_Type'] == 'prepaid' && \$web_subscription['active_Subscription'] == 1 && \$web_subscription['active_in_porta_billing'] == 0){ ?? <a title="create account with payment" target="iws_<?=\$web_subscription['i_web_subscription'] ?>" href="prepaid_account_creation.php?iws=<?=\$web_subscription['i_web_subscription'] ?>&action=info"> <?php } elseif (\$web_subscription['Account_Type'] != 'prepaid' && \$web_subscription['active_Subscription'] == 1 && \$web_subscription['active_in_porta_billing'] == 0) { ?? <a title="create account and upload contract" target="iws_<?=\$web_subscription['i_web_subscription'] ?>" href="contract_account_creation.php?iws=<?=\$web_subscription['i_web_subscription'] ?>&action=info"> <?php } elseif (\$web_subscription['active_Subscription'] == 0 && \$web_subscription['active_in_porta_billing'] == 1) { </pre>	The last cell is the action cell. We display a different link and icon in this cell according to the status of the account and the type of the account. For example if the subscription is pending and is for a prepaid account, we


```

        $web_subscription_dir = $ROOT_DIR . "/web_subscription_files/" .
$web_subscription['i_web_subscription'];

        $file = $web_subscription_dir . '/' .
$web_subscription['activation_file'];

        if ( isset($web_subscription['activation_file'] ) &&
$web_subscription['activation_file'] != '' && file_exists( $_SERVER['DOCUMENT_ROOT'] .
'/' . $file ) ){
            ?>
            <a title="look for saved payment/contract" target="_blank" href="<?=
$file ?>"></a>
            <?php
                }
            }
            ?>
        </td>

```

display the button for manually enter a payment that will open the account.

If it is for private or business, we display the button for manually upload a signed contract that will open the account.

If it is a processed subscription, we put a button for displaying the account or payment proof that have been uploaded.

```

        </tr>
        <?php } ?>
    </tbody>
</table>
</div>

```

```
<?php
```

End of the table.

```

?>

<div id="table-navigator">
  <?php if ( $page_to_display > $min_pages ){ ?>
    <a href="?page=<?= ( $page_to_display - 1 ) ?>">&lt; previous</a>
  <?php } ?>
  <?php
    $pages_select_offset = 2;
    $current_style = "font-weight:bold;color:black";
    for ($i=max($min_pages, $page_to_display - $pages_select_offset ); $i <
$page_to_display ; $i++) {
      $style = "";
      if ( $page_to_display == $i ) $style = $current_style;
      echo '<a href="?page=' . $i . '" style="' . $style . '">' . $i . '</a> ';
    }
    for ($i=$page_to_display; $i <= min($max_pages, $page_to_display +
$page_select_offset) ; $i++) {
      $style = "";
      if ( $page_to_display == $i ) $style = $current_style;
      echo '<a href="?page=' . $i . '" style="' . $style . '">' . $i . '</a> ';
    }
  ?>
  <?php if ( $page_to_display < $max_pages ){ ?>
    <a href="?page=<?= ( $page_to_display + 1 ) ?>">next &gt;</a>
  <?php } ?>
</div>

```

Navigation
though pages.

```

<div id="preferences-wrapper">

  <div id="preferences">

```

Beginning of the
preferences.

```

<h2>Columns to display:</h2>
<br><br>

```

Column selection.

We display all
column with a

```

<form action="#" method="POST">
<?php
    $checked_columns = array();
    foreach ($columns_to_display as $key => $value) {
        $checked_columns[$value] = 1;
    }
    foreach ($available_columns as $key => $value) {
        $selected='';
        if ( isset($checked_columns[$key]) ){
            $selected=' checked="on" ';
        }
        echo '<span class="column_select"><input name="display_columns[]"
value="'. $key. '" type="checkbox" '. $selected. '> ' . $key. '</input></span>';
    }
?>
<br><br>
<br><br>

```

checkbox for
selecting the one
we want.

```

<h2>
    Displaying preferences
</h2>
<br><br>
<table id="displaying_preferences">

```

Displaying
preferences.

```

<tr>
    <td>Number of row per pages :</td>
    <td>
        <select name="results_per_page">
            <?php
                $results_page = array('5', '10', '20', '50', '100');

                foreach ($results_page as $key => $value) {
                    $selected='';
                    if ( $value == $_SESSION['WS_TABLE_PARAMS']['results_per_page'] ){
                        $selected=' selected="selected" ';
                    }
                    echo '<option value="'. $value. '"
'. $selected. '>' . $value. '</option>';
                }
            ?>
        </select>
    </td>
</tr>

```

Selection of the
number of row
per pages.

<pre> </select> </td> </tr> </pre>	
<pre> <tr> <td>Order by :</td> <td> <select name="order_by"> <?php foreach (\$available_Columns as \$key => \$value) { \$selected=''; if (\$key == \$_SESSION['WS_TABLE_PARAMS']['order_by']){ \$selected=' selected="selected" '; } echo '<option value="'. \$key. '" ' . \$selected. '>' . \$key. '</option>'; } ?> </select> </td> </tr> </pre>	Selection of the column to order by.
<pre> <tr> <td>Order direction :</td> <td> <select name="order_direction"> <?php \$orders = array('ab' => 'smaller to greater', 'ba' => 'greater to smaller'); foreach (\$orders as \$key => \$value) { \$selected=''; if (\$key == \$_SESSION['WS_TABLE_PARAMS']['order_direction']){ \$selected=' selected="selected" '; } echo '<option value="'. \$key. '" ' . \$selected. '>' . \$value. '</option>'; } ?> </select> </td> </tr> </pre>	Selection of the direction of the order. - Smaller to greater - Or - Greater to smaller

<pre> <tr> <td>Account types to display :</td> <td> <?php foreach (\$available_Account_Types as \$key => \$value) { \$selected=''; if (isset(\$account_type_display[\$key])){ \$selected=' checked="on" '; } echo '<input name="account_type_display[]" value="'. \$key. '" type="checkbox" '. \$selected. '>'. \$value. '</input>'. "\n"; } ?> </td> </tr> <tr> <td>Status to display</td> </pre>	Type of account to display : - Business - Private - prepaid
<pre> <td> <?php foreach (\$available_Subscription_Statuses as \$key => \$value) { \$selected=''; if (isset(\$subscription_status_display[\$key])){ \$selected=' checked="on" '; } echo '<input name="subscription_status_display[]" value="'. \$key. '" type="checkbox" '. \$selected. '>'. \$value. '</input>'. "\n"; } ?> </td> </pre>	Status of the subscription: - Pending - Processed - Deleted
<pre> </tr> </table> </div> <div id="column-selector-submit" > </pre>	End of preferences and validation button

[1'e9q'1 auto signup]

```
<input type="hidden" name="action" value="new_pref" >
<input class="pure-button pure-button-primary" type="submit" value="Submit">
</div>
</form>
</div>
</div>
</body>
</html>
```

for submitting
the form.

```
<?php
$mysqli->close();
?>
```

Closing mysql
connection.

Page for opening prepaid account on manual payment

CODE

```
<?php

if(session_id() == '') {session_start();}

include_once
"classes/WebSubscriptions_PortaBilling_Management/WebSubscriptions_PortaBilling_Management.php";

include_once "credentials/soap.conf.php";

$SOAPCredentials = new SOAPCredentials;
$SOAP_user = $SOAPCredentials->get_user();
$SOAP_password = $SOAPCredentials->get_password();

include_once "classes/PortaBillingSoapClient.php";

?>
```

COMMENT

Session start

```
<!doctype html>
```

HTML header.

<pre> <html> <head> <meta http-equiv="X-UA-Compatible" content="IE=Edge;chrome=1" /> <meta charset="UTF-8"> <title>Web subscriptions management</title> <link rel="stylesheet" href="css/pure-min.css"> <link rel="stylesheet" href="css/styles.css"> <script src="js/jquery-1.11.1.min.js"></script> <!--[if lt IE 7]> <script src="http://ie7-js.googlecode.com/svn/version/2.1(beta4)/IE7.js"></script> <![endif]--> </head> </pre>	
<pre> <body> <div id="head" style="padding:15px; width:80%; margin: 0 auto;" > <!-- --> <h1 style="text-align:center;margin-top:20px;">Enter payment to create prepaid account</h1> </div> <center> <div id="container" style="width:500px; text-align:left; border:1px solid grey"> <form action="http://switzernet.com/3/support/141014-web-subscriptions/prepaid_account_creation.php" method="POST" enctype="multipart/form-data"> </pre>	Container of the form.
<pre> <?php if (isset(\$_GET['iws']) && preg_match('/^[0-9]+\$/', \$_GET['iws']) && \$_GET['action'] == 'info'){ \$wpbm = new WebSubscriptions_PortaBilling_Management(); \$wpbm->set_i_web_subscription(\$_GET['iws']); \$wpbm->get_web_subscription(); ?> </pre>	Getting parameters from the url.
<pre> <h2>Subscription info</h2>

 <?php \$i = 1; foreach (\$wpbm->web_subscription['subscribedNumbers'] as \$value) { echo 'Account '.\$i.': '. \$value['h323_id'].'

'; \$i++; } </pre>	Displaying basic information about the subscription.

<pre> \$wpbm->_close(); } ?> </form> </pre>	
<pre> <?php \$ERROR_MESSAGE = ""; // print_r(\$_SERVER); \$ROOT_DIR = \$_SERVER['DOCUMENT_ROOT'].'/'.dirname(\$_SERVER['SCRIPT_NAME']); if (isset(\$_POST['iws']) && preg_match('/^[0-9]+\$/', \$_POST['iws']) && \$_POST['action'] == 'create_and_pay'){ </pre>	If the action is to create the account and pay ...
<pre> \$creation = false; if (! isset(\$_POST['amount']) ! preg_match('/^[0-9]+(\.[0-9]{0,2})?\$/', \$_POST['amount'])){ \$ERROR_MESSAGE = "NO AMOUNT DEFINED"; </pre>	We verify the amount.
<pre> } else { \$wpbm = new WebSubscriptions_PortaBilling_Management(); \$wpbm->set_i_web_subscription(\$_POST['iws']); \$wpbm->get_web_subscription(); </pre>	We get the subscription from the database through the management class.
<pre> if (\$wpbm->web_subscription['Account_Type'] != 'pre'){ \$ERROR_MESSAGE = "This is not a prepaid account !"; </pre>	This is only for prepaid accounts. We stop display and error if it is not a prepaid account.

```

} else {
    $FILE_UPLOAD_ERROR = FALSE;

    if ( isset($_FILES['payment_file']) && $_FILES['payment_file']['size'] > 0 ){

        $FILE_UPLOAD_ERROR = TRUE;

        # file upload first.
        $extensions_valides = array( 'jpg' , 'jpeg' , 'gif' , 'png' , 'pdf' );
        $extension_upload = strtolower( substr( strrchr($_FILES['payment_file']['name'],
        '.'), 1) );

        if ($_FILES['payment_file']['error'] > 0){
            $ERROR_MESSAGE = "File transfert failed : upload error !\n";
        } elseif ( $_FILES['payment_file']['size'] > 1048576 ) {
            $ERROR_MESSAGE .= "File transfert failed : file to big\n";
        } elseif ( !in_array($extension_upload,$extensions_valides) ) {
            $ERROR_MESSAGE .= "File transfert failed : bad extension file\n";
        } else {

            $web_subscription_dir = $ROOT_DIR . "/web_subscription_files/" . $wpbm-
            >web_subscription['i_web_subscription'];

            if ( !file_exists( $web_subscription_dir) ){
                mkdir( $web_subscription_dir, 0755, true );
            }

            $filename = time().'-payment.'.$extension_upload;
            $web_subscription_file = $web_subscription_dir . '/' . $filename;

            $resultat =
            move_uploaded_file($_FILES['payment_file']['tmp_name'],$web_subscription_file);

            if ($resultat){

```

We upload
the payment
proof if any.

If upload is
ok, we update

<pre> \$FILE_UPLOAD_ERROR = FALSE; \$wpbm->update_web_subscription_active_file(\$filename); } } } </pre>	the database for retrieving the document corresponding to this subscription latter.
<pre> if (\$FILE_UPLOAD_ERROR == FALSE){ \$creation = \$wpbm->create_customer_and_account_from_web_subscription(); </pre>	We create the customer.
<pre> if (\$creation == TRUE){ \$wpbm->send_creation_notification("prepaid"); </pre>	Sending the notification to customer.
<pre> \$wpbm->get_web_subscription(); \$ServiceCustomer = new PortaBillingSoapClient('https://slave.switzernet.com:8444', 'Admin', 'Customer'); \$session_id = \$ServiceCustomer->_login(\$SOAP_user, \$SOAP_password); \$ServiceCustomer->_setSessionId(\$session_id); \$MakeCustomerTransactionRequest = array('i_customer' => \$wpbm->web_subscription['pb_i_customer'], 'visible_comment' => 'paiement', 'internal_comment' => 'paiement', 'action' => 'Manual payment', 'amount' => \$_POST['amount'], 'suppress_notification' => 0, 'h323_conf_id' => ''); try{ // do the transaction \$MakeCustomerTransactionResponse = \$ServiceCustomer- >make_transaction(\$MakeCustomerTransactionRequest); </pre>	Processing the payment.

<pre> } catch (SoapFault \$fault) { \$ERROR_MESSAGE = "Error during payment !\n\n".\$fault; } \$ServiceCustomer->_logout(); } } \$wpdb->_close(); } </pre>	
<pre> ?> <?php if (\$ERROR_MESSAGE == ""){ ?> <h2>Account creation info</h2>

 Account has been created and payment has been added in billing.

 <button class="pure-button" type="button" onclick="window.opener.location.reload(true);window.close()">Close</button> </pre>	Account creation and payment is done.
<pre> <?php } else { ?> <h2>ERROR</h2>

 <?= \$ERROR_MESSAGE ?>

 <button class="pure-button" type="button" onclick="window.opener.location.reload(true);window.close()">Close</button> </pre>	Displaying the error message if any.
<pre> <?php } } ?> </div> </center> </pre>	End of script.

```
</body>
</html>
```

Page for opening account on signed contract reception

CODE	COMMENT
<pre><?php if(session_id() == '') {session_start();}</pre>	Session start
<pre><?php include_once "classes/WebSubscriptions_PortaBilling_Management/WebSubscriptions_PortaBilling_Management.php"; include_once "credentials/soap.conf.php"; \$SOAPCredentials = new SOAPCredentials; \$SOAP_user = \$SOAPCredentials->get_user(); \$SOAP_password = \$SOAPCredentials->get_password(); include_once "classes/PortaBillingSoapClient.php"; ?></pre>	Includes.
<pre><!doctype html> <html> <head> <meta http-equiv="X-UA-Compatible" content="IE=Edge;chrome=1" /> <meta charset="UTF-8"> <title>Web subscriptions management</title> <link rel="stylesheet" href="css/pure-min.css"> <link rel="stylesheet" href="css/styles.css"> <script src="js/jquery-1.11.1.min.js"></script> <!--[if lt IE 7]></pre>	HTML header.

<pre> <script src="http://ie7-js.googlecode.com/svn/version/2.1(beta4)/IE7.js"></script> <![endif]--> </head> </pre>	
<pre> <body> <div id="head" style="padding:15px; width:80%; margin: 0 auto;" > <!-- --> <h1 style="text-align:center;margin-top:20px;">Create customer account with signed contract</h1> </div> <center> <div id="container" style="width:500px; text-align:left; border:1px solid grey"> <form action="http://switzernet.com/3/support/141014-web-subscriptions/contract_account_creation.php" method="POST" enctype="multipart/form-data"> <?php if (isset(\$_GET['iws']) && preg_match('/^[0-9]+\$/', \$_GET['iws']) && \$_GET['action'] == 'info'){ \$wpbm = new WebSubscriptions_PortaBilling_Management(); \$wpbm->set_i_web_subscription(\$_GET['iws']); \$wpbm->get_web_subscription(); //print_r(\$wpbm); ?> </pre>	Getting parameters from the url.
<pre> <h2>Subscription info</h2>

 <?php \$i = 1; foreach (\$wpbm->web_subscription['subscribedNumbers'] as \$value) { echo 'Account '.\$i.': '. \$value['h323_id'].'

'; \$i++; } echo 'Firstname : '. \$wpbm->web_subscription['Firstname'].'

'; echo 'Lastname : '. \$wpbm->web_subscription['Lastname'].'

'; ?> <input type="hidden" name="MAX_FILE_SIZE" value="1048576" /> Upload signed contract : <input name="contract_file" type="file" required>
 <div style="text-align:center; width:100%"> <input type="hidden" name="iws" value="<?= \$wpbm->web_subscription['i_web_subscription'] ?>" > </pre>	Displaying basic information about the subscription.

<pre> if (\$wpbm->web_subscription['Account_Type'] == 'pre'){ \$ERROR_MESSAGE = "This is a prepaid account !"; </pre>	The account must not be prepaid.
<pre> } else { # file upload first. \$extensions_valides = array('jpg' , 'jpeg' , 'gif' , 'png' , 'pdf'); \$extension_upload = strtolower(substr(strrchr(\$_FILES['contract_file']['name'], '.')) ,1)); if (!isset(\$_FILES['contract_file'])){ \$ERROR_MESSAGE = "File transfert failed : no file defined !\n"; } elseif (\$_FILES['contract_file']['error'] > 0){ \$ERROR_MESSAGE = "File transfert failed : upload error !\n"; } elseif (\$_FILES['contract_file']['size'] > 1048576) { \$ERROR_MESSAGE .= "File transfert failed : file to big\n"; } elseif (!in_array(\$extension_upload,\$extensions_valides)) { \$ERROR_MESSAGE .= "File transfert failed : bad extension file\n"; } else { \$web_subscription_dir = \$ROOT_DIR . "/web_subscription_files/" . \$wpbm->web_subscription['i_web_subscription']; if (!file_exists(\$web_subscription_dir)){ mkdir(\$web_subscription_dir, 0755, true); } \$filename = time().'-signed-contract.'.\$extension_upload; \$web_subscription_file = \$web_subscription_dir . '/' . \$filename; \$resultat = move_uploaded_file(\$_FILES['contract_file']['tmp_name'],\$web_subscription_file); </pre>	Uploading the contract and saving.
<pre> if (\$resultat){ \$FILE_UPLOADED = TRUE; </pre>	If the upload succeed we

<pre> \$wpbm->update_web_subscription_active_file(\$filename); } </pre>	update the database.
<pre> } if (\$FILE_UPLOADED){ \$creation = \$wpbm->create_customer_and_account_from_web_subscription(); } </pre>	Then we create the customer in the billing.
<pre> if (\$creation == TRUE){ \$wpbm->send_creation_notification("contract"); } } </pre>	And finally send the account creation notification.
<pre> } } \$wpbm->_close(); } ?> </pre>	Closing the web subscription instance.
<pre> <?php if (\$ERROR_MESSAGE == ""){ ?> <h2>Account creation info</h2>

 Account has been created. Please verify in porta-billing.

 </pre>	Informing the account has been created.
<pre> <button class="pure-button" type="button" onclick="window.opener.location.reload(true);window.close()">Close</button> </pre>	Button for closing the windows.
<pre> <?php } else { ?> <h2>ERROR</h2>

 <?= \$ERROR_MESSAGE ?> </pre>	Displaying the error is the account creation failed.

<pre>

</pre>	
<pre> <button class="pure-button" type="button" onClick="window.opener.location.reload(true);window.close()">Close</button> <?php } ?> </div></pre>	Button for closing the windows.
<pre></center> </body> </html></pre>	End of the page.

Liens

Ce document :

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

* * *



Copyright © 2014 by Switzernet