automatic account creation
 on e-sign and e-pay

Document created on [2014-10-15](http://switzernet.com/3/public/141015-automatic-web-subscriptions/index1.docx)
Nicolas Bondier

[[pdf](index.pdf)][[doc](index.docx)][[htm](index.htm)]

\* \* \*



Copyright © 2014 by Switzernet

Contents

[Introduction 3](#_Toc401592671)

[Modifications on subscriptions web pages 4](#_Toc401592672)

[Email validation page 4](#_Toc401592673)

[Final subscription page (partial) 13](#_Toc401592674)

[Script for managing web subscription and porta-gilling accounts 19](#_Toc401592675)

[Adding porta-billing account creation to payment notifications pages 40](#_Toc401592676)

[PayPal notification page (partial) 41](#_Toc401592677)

[PostFinance notification page (partial) 42](#_Toc401592678)

[Pages for manually enter payment and automatic account creation 43](#_Toc401592679)

[Main table page 43](#_Toc401592680)

[Page for opening prepaid account on manual payment 57](#_Toc401592681)

[Page for opening account on signed contract reception 63](#_Toc401592682)

[Liens 70](#_Toc401592683)

# 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 |
| <?php **if**(session\_id() == '') {session\_start();} | Session start  |
| $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** $\_SERVER['DOCUMENT\_ROOT']."/public/070608-subscribe/include/calcul\_price.php"; | Including all necessary files. |
| // DEFAULT LANGUAGE**if** (!isset($\_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'; }}**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"; | Language file selctor. |
| $mysqli = **new** mysqli(DB\_SERVER,DB\_SERVER\_USERNAME,DB\_SERVER\_PASSWORD,DB\_DATABASE);**if** ($mysqli->connect\_errno) { **echo** "Echec lors de la connexion à MySQL : (" . $mysqli->connect\_errno . ") " . $mysqli->connect\_error;} | Opening MySQL connection. |
| **if** ( isset($\_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` ) VALUES ( NOW(),?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?)";  | The request for inserting the new subscription in MySQL. |
| $Email\_Code = freeRandomString(8); | Generating the Email validation code. |
|  **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**;  } **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]); $\_POST['numero1'] = convert\_to\_41XXXXXXXXX($\_POST['numero1']); $\_POST['numero2'] = convert\_to\_41XXXXXXXXX($\_POST['numero2']); $Paper\_Invoice = 0; **if** ( $\_POST['type\_facture'] == 'type\_facture\_poste' ){ $Paper\_Invoice = 1; } | Verifying all filed before inserting in MySQL database. |
|  $stmt = $mysqli->prepare($sql); **if** ( **false**===$stmt ) { **die**('An error occured'); } $stmt->bind\_param('sssssssssssssssssssi', $\_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'], $\_POST['Fax'], $\_POST['eMail'], $\_POST['Parrain'], $\_POST['promo\_code'], $\_POST['Comment'], $Email\_Code, $Paper\_Invoice ); **if** ( **false**===$stmt ) { **die**('An error occured'); } $stmt->execute(); **if** ( **false**===$stmt ) { **die**('An error occured'); } $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(); } } | Inserting the customer and the selected numbers in the MySQL database. |
|  $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' ) { $cher = TRAITEMENT\_WELCOME\_BONJOURE; } //$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;"> <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($Email\_Code).'"> <span style=" cursor: pointer; background-color: #0078e7; color: #fff; font-size: 100%; padding: .5em 1.5em; border: 0 rgba(0,0,0,0); text-decoration: none; border-radius: 2px; -webkit-transition: .1s linear -webkit-box-shadow; -moz-transition: .1s linear -moz-box-shadow; -ms-transition: .1s linear box-shadow; -o-transition: .1s linear box-shadow; transition: .1s linear box-shadow; display: inline-block; zoom: 1; line-height: normal; white-space: nowrap; vertical-align: baseline; text-align: center; -webkit-user-drag: none; -webkit-user-select: none; -moz-user-select: none; -ms-user-select: none; user-select: none;"> '. EMAIL\_VALIDATION\_EMAIL\_BUTTON .' </span> </a> </td> </tr> <tr style="height:40px"> <td style="text-align: center;vertical-align: top;"> <p>'. EMAIL\_VALIDATION\_EMAIL\_TEXT\_2 .'</p> </td> </tr> <tr style="height:40px"> <td style="text-align: left;vertical-align: top;"> <p>'. EMAIL\_VALIDATION\_EMAIL\_TEXT\_3 .' <a style="color: #0078e7; text-decoration: none;" href="http://switzernet.com/email\_validation">http://switzernet.com/email\_validation</a> '. EMAIL\_VALIDATION\_EMAIL\_TEXT\_4 .' '.strrev( implode('-', str\_split(strrev($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://switzernet.com" >www.switzernet.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**;} | Sending email to the new customer with the validation code to enter. |
| } |  |
| $mysqli->close();?> | Closing MySQL connection. |
| <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; "> <br> <center> <h1><?= INSCR\_TITLE ?></h1> </center> <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> <input class="input\_form" type="text" name="emailValidationCode" maxlength="20" value="" size="30"> <?php **if** ( isset($\_POST['eMail']) && $\_POST['eMail'] != '' && $\_POST['eMail'] != **NULL** ){ ?> <input class="input\_form" type="hidden" name="eMail" maxlength="20" value="<?= $\_POST['eMail'] ?>" size="20"> <?php } ?> </td> </tr> </table> <br><br> <input type="button" class="pure-button" value="<?= BACK ?>" onClick=\"return retour(formul)\">&nbsp; &nbsp; &nbsp; &nbsp; &nbsp; &nbsp; <input class="pure-button pure-button-primary" type="submit" value="<?= VALID ?>" onClick="suite(formul)"> </form> </td> </tr> <tr> <td> <? **include\_once** "../060801-web/includes/footer.php" ?> </td> </tr> </table></center></body></html> | Form for entering the email code. |
| <?php**function** generateRandomString($length = 5) { $characters = '0123456789abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ'; $characters = '0123456789ABCDEFGHIJKLMNOPQRSTUVWXYZ'; $randomString = ''; **for** ($i = 0; $i < $length; $i++) { $randomString .= $characters[rand(0, strlen($characters) - 1)]; } **return** $randomString;} | Generating random string password. |
| **function** freeRandomString($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 |
| <?php **if**(session\_id() == '') {session\_start();} | Session start  |
| $debugMode=**False**; // Put False to check usual mode for debug user$page\_title = "Finished Subscribe"; | Starting the session |
| $ERROR\_MESSAGE = '';$ERROR\_TITLE = '';$ERROR = **FALSE**;?> | Preparing error variables. |
| <html><head><!-- Piwik : Do not remove --> <script type="text/javascript">var pkBaseURL = (("https:" == document.location.protocol) ? "https://switzernet.com/3/public/120427-site-stats/" : "http://switzernet.com/3/public/120427-site-stats/");document.write(unescape("%3Cscript src='" + pkBaseURL + "piwik.js' type='text/javascript'%3E%3C/script%3E"));</script><script type="text/javascript">try { var piwikTracker = Piwik.getTracker(pkBaseURL + "piwik.php", 1); piwikTracker.setDocumentTitle("Subscribe end"); piwikTracker.trackPageView(); piwikTracker.enableLinkTracking();} catch( err ) {}</script><noscript><p><img src="http://switzernet.com/3/public/120427-site-stats/piwik.php?idsite=1" style="border:0" alt="" /></p></noscript><!-- End Piwik Tracking Code --> | Javascript code for piwik. |
| <?php$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 = "";$eMail = "";$Parrain = "";$comment = "";$type\_facture = ""; | Preparing the variables of the subscription. |
| **if** ( ( !isset( $\_POST['emailValidationCode']) || !isset( $\_POST['eMail'] ) ) && ( !isset($\_GET['em']) || !isset($\_GET['cn']) ) ){ $ERROR = "WRONG\_PAGE"; | Checking the POST and GET parameters.Creating a new error if these fields are not present. |
| } **else** { **if** ( isset($\_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. "<br>"; //echo "validationCode : ".$validationCode. "<br>"; $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(); | 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. |
|  **if** ( $result < 1 ){ $ERROR = "WRONG\_VALIDATION\_CODE"; } **else** { **if** ( $verified\_Email == 1 ){ $ERROR = "HAS\_BEEN\_VERIFIED\_BEFORE"; } **else** { | Checking ifwe have a result. |
| # [...] not commented |  |
| **function** update\_internal\_subscription\_mail\_id($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();  $stmt->close(); } | Function for updating the web subscription record with the message id used to send the email to internal mailbox. |
|  **function** update\_customer\_subscription\_mail\_id($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();  $stmt->close(); } | Function for updating the web subscription record with the message id used to send the email to customer mailboxes. |
|  **function** update\_esign\_code($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();  $stmt->close(); } | Function for updating the web subscription record with a new esign code. |
|  **function** update\_i\_address\_validation($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();  $stmt->close(); } | Function for updating the web subscription with the id of the record inserted in the addresse validation table. |
|  **function** update\_web\_login($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();  $stmt->close(); } | Function to update the web login in subscription DB. |
|  **function** update\_web\_password($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();  $stmt->close(); } | Function to update the web password in subscription DB. |
|  **function** update\_h323\_password($i\_web\_subscription\_reserved\_number, $h323\_password){ **if** ( ! preg\_match('/^[0-9]+$/', $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();  $stmt->close(); } | Function to update the VoIP password password in subscription DB. |
|  **function** update\_reserved\_numbers\_i\_web\_subscription($number,$i\_web\_subscription){ $number = convert\_to\_41XXXXXXXXX($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; $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(); } | Function to update total amount of the order in subscription table. |

# Script for managing web subscription and porta-gilling accounts

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

|  |  |
| --- | --- |
| CODE | COMMENT |
| <?php**class** **WebSubscriptions\_PortaBilling\_Management**{ | Getting information posted to this page. |
|  **private** $ServiceAccount; **private** $ServiceCustomer; **private** $LocalMysqli; **private** $MasterMysqli; | Declaration to the private properties. These are the connections to MySQL and SOAP. |
|  **public** $web\_subscription; **public** $customer\_info = **array**(); **public** $account\_list = **array**(); | Declaring the public properties that we will access later.$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. |
|  **function** \_\_construct(){ $this->soap\_connect(); $this->web\_db\_connect(); $this->master\_db\_connect(); } | We initiate the connections to MySQL and SOAP in the class constructor.  |
| **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; $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 !"); } } | Connection to SOAP, Account and customer services. |
| **private** **function** web\_db\_connect(){ $filename = 'connections/local.mysqli.class.php'; **if**(!@**include\_once**( $filename ) ) **throw** **new** WebSubscriptions\_PortaBilling\_Management\_Exception( "Failed to include ". $filename ); **try** { $LocalMysql = **new** LocalMysqli; $this->LocalMysqli = $LocalMysql->connection; **return** $this->LocalMysqli; } **catch** (mysqli\_sql\_exception $e) { **throw** **new** WebSubscriptions\_PortaBilling\_Management\_Exception("Could not connect to mysql server !".$e); } } | Connection to the local database. |
| **private** **function** master\_db\_connect(){ $filename = 'connections/master.mysqli.class.php'; **if**(!@**include\_once**( $filename ) ) **throw** **new** WebSubscriptions\_PortaBilling\_Management\_Exception( "Failed to include ". $filename ); **try** { $MasterMysqli = **new** MasterMysqli; $this->MasterMysqli = $MasterMysqli->connection; **return** $this->MasterMysqli; } **catch** (mysqli\_sql\_exception $e) { **throw** **new** WebSubscriptions\_PortaBilling\_Management\_Exception("Could not connect to mysql server !".$e); } } | Connection to the master database. |
|  # 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**; } | 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. |
|  **public** **function** create\_customer\_and\_account\_from\_web\_subscription(){ $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**; } | The main function to call for creating the customer and its accounts.It also update the local database when the subscription is done. |
|  **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**; } | Check if the account exists in the porta-billing. |
|  **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 ); $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**; } | Check if the customer exists in porta-billing. The name and login matters because there are unique. |
|  **public** **function** get\_web\_subscription(){ $subscription = **array**(); $sql = "SELECT \* FROM Web\_Subscriptions ws WHERE ws.i\_web\_subscription = ?"; $stmt = $this->LocalMysqli->prepare($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 = $this->LocalMysqli->prepare($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; } } $stmt->close(); $subscription['subscribedNumbers'] = **array**(); **foreach** ($tmp **as** $key => $value) { array\_push($subscription['subscribedNumbers'], $value); } $this->web\_subscription = $subscription;  **return** $subscription; } | Getting the web subscriptions from the local website and also the subscribed numbers. |
|  **public** **function** verify\_esign\_code( $code ){ $this->get\_web\_subscription(); **if** ( $this->web\_subscription['Esign\_Code'] == $code ){ **return** **true**; } **return** **false**; } | Function to check if the esign code is valid. |
|  **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**; } } | 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'; $mail->AddAddress($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\_XXXX = ''; $XXXXXXXXXXX = ''; //print\_r($this); **foreach** ( $this->web\_subscription[ 'subscribedNumbers' ] **as** $account ) { //if ( $this->customer\_exists\_in\_pb( NULL, $account['h323\_id'] ) ){ $XXX\_XXX\_XXXX = $this->convert\_to\_0XX\_XXX\_XXXX( $account['h323\_id'] ); $XXXXXXXXXXX = $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'] ); $subscrition\_date = date('Y-m-d',$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, '[[41XXXXXXXXX]]' => $XXXXXXXXXXX, '[[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 " . $subscrition\_date; **if** ( $validation\_type == "postal\_validation" ){ $mail\_Subject = "[Signed postal] " . $XXX\_XXX\_XXXX . " on " . $subscrition\_date; $mail\_Body .= RECEIVED\_POSTAL\_VALIDATION; } **elseif** ( $validation\_type == "prepaid" ){ $mail\_Subject = "[Signed prepaid] VOIP-" . $XXX\_XXX\_XXXX . " on " . $subscrition\_date; $mail\_Body .= RECEIVED\_PAYMENT\_PREPAY;  } **elseif** ( $validation\_type == "esign" ){ $mail\_Subject = "[Signed esign] VOIP-" . $XXX\_XXX\_XXXX . " on " . $subscrition\_date; $mail\_Body .= RECEIVED\_CONTRACT\_ESIGN;  } **elseif** ( $validation\_type == "contract" ){ $mail\_Subject = "[Signed contract] VOIP-" . $XXX\_XXX\_XXXX . " on " . $subscrition\_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\_swubscription['Account\_Type'] != 'pre' ){ $mail\_Body .= INFO\_CREDIT\_LIMIT; } $mail\_Body .= IMPORTANT\_POINTS; **if** ( $this->web\_swubscription['Account\_Type'] != 'pre' ){ $mail\_Body .= IMPORTANT\_POINT\_INVOICES; } **if** ( $this->web\_swubscription['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\_swubscription['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 style="width:150px" src="'.$this->base64\_encode\_image( dirname(**\_\_FILE\_\_**) . "/img/switzernet.gif","gif").'" alt="switzernet"></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(); } | 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. |
|  **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(); } | Function that call all needed functions for creating the customer. |
|  **private** **function** create\_accounts\_from\_web\_subscription(){ $this->get\_web\_subscription(); $this->create\_accounts\_list\_from\_web\_subscription(); $this->fix\_accounts\_info(); $this->create\_accounts\_in\_pb(); } | Function that call all needed functions for creating the accounts. |
|  **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(); } | Function for updating the flag when the subscription has been added to porta-billing. |
|  **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^^^^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 ); } } | 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.  |
|  **private** **function** fix\_customer\_info(){ $sql = "UPDATE Customers SET service\_flags = 'NNANNNPN 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(); } | After the customer has been created in the billing, the value of service\_flags is incorrect. We update this with MySQL in the billing.  |
|  **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(); } | After the account has been created in the billing, the value of service\_flags is incorrect. We update this with MySQL in the billing. |
|  **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(); } | This function create a customer notepad for the customer. |
|  **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); $state = $stmt->bind\_param( "i",$this->customer\_info['i\_customer'] ); $stmt->execute(); $stmt->close(); } | This function write to the customer notepad the stamps we were manually inserting before. |
|  **private** **function** create\_customer\_info\_from\_web\_subscription(){ **if**(!@**include**( "/var/www/public/141005-subscribe/includes/currency\_prices.php" ) ) **throw** **new** WebSubscriptions\_PortaBilling\_Management\_Exception( "Failed to include /var/www//public/141005-subscribe/includes/currency\_prices.php" ); # name **if** ( isset($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->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; } | This function transform the web subscription into customer info that we will use with SOAP for creating the customer. |
|  **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; } | This function get the customer info we created and create the customer in the porta-billing with SOAP. |
|  **private** **function** charge\_total\_order(){ **if** ( $this->web\_subscription['Total\_order'] > 0 ){ $this->charge\_customer( $this->web\_subscription['Total\_order'], 'Total commande', 'subscription order' ); } } | This function get the amount of the order and pass the amount to the function to charge the customer. |
|  **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; } | 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 subscrptions $i\_subscription = ''; **if** ($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'] );  $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**;  } | 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. |
|  **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. |
|  **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**; } | This apply the update of the credit limit to the customer in porta-billing. |
|  **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(); } | This function updates the local MySQL database with the new i\_customer of the customer inserted in the porta-billing. |
|  **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(); } | Function for disabling a valid web subscription. |
|  **private** **function** create\_accounts\_in\_pb(){ $AddAccountResponses = **array**(); **foreach** ($this->account\_list **as** $account\_info ) { $AddAccountRequest = **array**( 'account\_info' => $account\_info ); **try** { $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** $AddAccountResponse; } | This function create the account in porta-billing with SOAP. |
|  **private** **function** stmt\_bind\_assoc (&$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); } | Function to build an array with MySQL results. |
|  **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** $res['i\_web\_subscription']; } **return** **false**; } | Getting a web subscription in the local database from an ordered account id. |
|  **private** **function** print\_r\_this\_true(){ **return** print\_r($this, **TRUE**); } |  Function for printing all class data. Used for debugging. |
|  **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** $number; } **else** { **return** **FALSE**; } } | Function for converting any number string to the following format 0XX XXX XXX. |
|  **private** **function** convert\_to\_41XXXXXXXXX($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**; } } | Function for converting any number string to the following format 41XX XXX XXX. |
|  # Setters and getters **public** **function** get\_all\_variables(){ **return** $this; } | Function for returning the whole class. |
|  **public** **function** set\_i\_web\_subscription($i\_web\_subscription){ $this->web\_subscription['i\_web\_subscription'] = $i\_web\_subscription; } | Function for selecting the web subscription. |
|  **private** **function** set\_web\_subscription($web\_subscription){ $this->web\_subscription = $web\_subscription; } | Function for loading the web subscription data. |
|  **private** **function** set\_customer\_info($customer\_info){ $this->customer\_info = $customer\_info; } | Function for setting the customer info (format) to the current class. |
|  **private** **function** set\_account\_list($account\_list){ $this->account\_list = $account\_list; } | Function for setting the account list to the current class. |
|  # Closing functions **public** **function** \_close(){ $this->LocalMysqli->close(); $this->MasterMysqli->close(); $this->\_\_destruct(); } | Function to call to close the the class. |
|  **function** \_\_destruct(){ $this->ServiceCustomer->\_logout(); $this->ServiceAccount->\_logout(); } | Destructor, which is called everytime when the class instance is closed. |
| } | End of the class. |
| **class** **WebSubscriptions\_PortaBilling\_Management\_Exception** **extends** Exception{ **public** **function** \_\_construct($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"; } }}?> | Extended exception class that also send an email when a problem occurs. |
|  |  |

#

# Adding porta-billing account creation to payment notifications pages

## PayPal notification page (partial)

|  |  |
| --- | --- |
| CODE | COMMENT |
| **include\_once** "classes/WebSubscriptions\_PortaBilling\_Management/WebSubscriptions\_PortaBilling\_Management.php"; | Session start  |
| [...]**if** ( $payment\_status == 'Completed' ){ | This is when the payment is ok. Account creation comes after. |
| $wpbm = **new** WebSubscriptions\_PortaBilling\_Management(); **if** ( isset($\_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() ) ); } } } } | 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 |
| **include\_once** "classes/WebSubscriptions\_PortaBilling\_Management/WebSubscriptions\_PortaBilling\_Management.php"; | Session start  |
| [...]**if** ( $\_POST['STATUS'] == '9' ){ | This is when the payment is ok. Account creation comes after. |
|  $wpbm = **new** WebSubscriptions\_PortaBilling\_Management(); **if** ( isset($\_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]; } $item\_number = format\_account($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() ) ); } } } } | 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. |

# Pages for manually enter payment and automatic account creation

## Main table page

|  |  |
| --- | --- |
| CODE | COMMENT |
| <?php **if**(session\_id() == '') {session\_start();} | 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'  ); $\_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'; | Definition of default values in session if the session does not exists. |
| **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']));} | 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). |
| **require\_once** 'connections/local.mysqli.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()) { $available\_Columns[$row['COLUMN\_NAME']] = 1; }}$available\_Columns['Accounts'] = 1; | Selecting column names of the table that will be displayed. |
| 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'] ); | Removing sensible data from the columns to display. |
| $available\_Account\_Types = **array**( 'pre' => 'Prepaid', 'priv' => 'Private', 'com' => 'Business');$available\_Subscription\_Statuses = **array**( 'pending' => 'Pending', 'processed' => 'Processed', 'deleted' => 'Deleted'); | Definition of the available accounts types and subscription statuses. |
| # Intitiate varibles$display\_columns = **array**();$page\_to\_display = 1;$results\_per\_page = 10;$order\_by = 'Subscription\_Timestamp';$order\_direction = 'DESC';$account\_type\_display = **array**();$subscription\_status\_display = **array**(); | 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'] ) ){ **foreach** ( $\_SESSION['WS\_TABLE\_PARAMS']['subscription\_status\_display'] **as** $value) { **if** ( isset($available\_Subscription\_Statuses[$value]) ){ $subscription\_status\_display[$value] = 1; } } }} | If action is to save new displaying preferences, we read the session variables. |
| $columns\_to\_display = **array**();**foreach** ($display\_columns **as** $name) { **if** ( isset( $available\_Columns[$name] ) ){ array\_push($columns\_to\_display, $name); }} | Removing columns that are not available. |
| # order by and order direction$order\_string = " ORDER BY wb.$order\_by $order\_direction "; | The order string construction. |
| # limit results per page$offset = ($page\_to\_display - 1) \* $results\_per\_page;$limit\_offset\_string = " LIMIT $results\_per\_page OFFSET $offset"; | We limit results per page with the user settings and we set the offset from which record we have to display result, which depends of the current page. |
| # filter account type$conditions = **array**(); | Creating an array the will contain all condition string for building the MySQL request. |
| **foreach** ($available\_Account\_Types **as** $key => $value) { **if** ( !isset( $account\_type\_display[$key] ) ){ array\_push($conditions, "Account\_Type != '".$key."'"); }} | 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. |
| array\_push($conditions, "verified\_Email = 1"); | Email code must be verified. |
| $subscription\_status\_display\_string = '';$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.")"); } } | 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 contatenate 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 AccountsFROM Web\_Subscriptions wb INNER JOIN Web\_Subscription\_Selected\_Numbers wssn ON wb.i\_web\_subscription = wssn.i\_web\_subscriptionWHERE $conditions\_string GROUP BY wb.i\_web\_subscription $order\_string $limit\_offset\_string";$web\_subscriptions = **array**();$account\_type\_converter = **array**('com' => 'business', 'priv' => 'private', 'pre' => 'prepaid');**if** ($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); }} | Selecting the matching subscriptions. |
| //echo $sql;$ROOT\_DIR = dirname($\_SERVER['SCRIPT\_NAME']);?> | Definition of the root directory. We need it to retieve documents on web server. |
| <!doctype html><html><head> <meta http-equiv="X-UA-Compatible" content="IE=Edge;chrome=1" /> <meta http-equiv="Content-Type" content="text/html; charset=utf-8"> <title>Web subscriptions management</title> <link rel="stylesheet" href="css/pure-min.css"> <link rel="stylesheet" href="css/styles.css"> <script src="js/jquery-1.11.1.min.js"></script> <!--[if lt IE 7]> <script src="http://ie7-js.googlecode.com/svn/version/2.1(beta4)/IE7.js"></script> <![endif]--></head><body> <div id="head" style="padding:15px; width:80%; margin: 0 auto;" > <!-- <img src="/public/060801-web/images/switzernet.gif" style="width:100px" alt="Switzernet"> --> <h1 style="text-align:center;margin-top:20px;">Web subscriptions management</h1> </div> <div id="container"> <div id="table-wrapper"> | HTML heading. |
|  <table id="table" class="pure-table"> <thead> <tr> <?php **foreach** ($columns\_to\_display **as** $value) { ?> <th name="<?= $value ?>" title="<?= $value ?>"><?= $value ?></th> <?php } ?> <th name="" title="">Actions</th> </tr> </thead> | Table header. We display the selected column fields. |
|  <tbody> <?php **foreach** ($web\_subscriptions **as** $web\_subscription) { ?> <tr> <?php **foreach** ($columns\_to\_display **as** $column\_name) { ?> <td><?= $web\_subscription[$column\_name] ?></td> <?php } ?> | Then for each subscription, we display the values. |
|  <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"><img style="width:30px;" src="img/add\_user\_and\_payment.fw.png"></a>  <?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"><img style="width:30px;" src="img/add\_user\_and\_contract.fw.png"></a> <?php } **elseif** ( $web\_subscription['active\_Subscription'] == 0 && $web\_subscription['active\_in\_porta\_billing'] == 1 ) {   $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 ?>"><img style="width:30px;" src="img/look-contract-payment.png"></a> <?php } }  ?> </td> | 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 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 + $pages\_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> <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> | Column selection.We display all column with a 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. |
|  <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> | Selection of the column to order by. |
|  <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> | Selection of the direction of the order.- Smaller to greaterOr- Greater to smaller |
|  <tr> <td>Account types to display :</td> <td> <?php **foreach** ($available\_Account\_Types **as** $key => $value) { $selected=''; **if** ( isset($account\_type\_display[$key]) ){ $selected=' checked="on" '; } **echo** '<span class="account\_type\_select"><input name="account\_type\_display[]" value="'.$key.'" type="checkbox" '.$selected.'>'.$value.'</input></span>'."\n"; } ?> </td> </tr> <tr> <td>Status to display</td> | Type of account to display :- Business- Private- prepaid |
|  <td> <?php **foreach** ($available\_Subscription\_Statuses **as** $key => $value) { $selected=''; **if** ( isset($subscription\_status\_display[$key]) ){ $selected=' checked="on" '; } **echo** '<span class="subscription\_status\_select"><input name="subscription\_status\_display[]" value="'.$key.'" type="checkbox" '.$selected.'>'.$value.'</input></span>'."\n"; } ?> </td> | Status of the subscription:- Pending- Processed- Deleted |
|  </tr> </table> </div> <div id="column-selector-submit" > <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> | End of preferences and validation button for submitting the form. |
| <?php$mysqli->close();?> | Closing mysql connection. |

## Page for opening prepaid account on manual payment

|  |  |
| --- | --- |
| CODE | COMMENT |
| <?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";?> | Session start  |
| <!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]> <script src="http://ie7-js.googlecode.com/svn/version/2.1(beta4)/IE7.js"></script> <![endif]--></head> | HTML header. |
| <body> <div id="head" style="padding:15px; width:80%; margin: 0 auto;" > <!-- <img src="/public/060801-web/images/switzernet.gif" style="width:100px" alt="Switzernet"> --> <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"> | Container of the form. |
|  <?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(); ?> | Getting parameters from the url. |
|  <h2>Subscription info</h2><br><br> <?php $i = 1; **foreach** ($wpbm->web\_subscription['subscribedNumbers'] **as** $value) { **echo** 'Account '.$i.': '. $value['h323\_id'].'<br><br>'; $i++; } **echo** 'Firstname : '. $wpbm->web\_subscription['Firstname'].'<br><br>'; **echo** 'Lastname : '. $wpbm->web\_subscription['Lastname'].'<br><br>'; ?> | Displaying basic information about the subscription. |
|  Amount paid by customer : <input name="amount" type="text" required value=""> <select name="Currency"> <?php $Currencies = **array**( 'CHF', 'USD', 'EUR' ); **foreach** ($Currencies **as** $value) { $selected=''; **if** ( $value == $wpbm->web\_subscription['Currency'] ){ $selected=' selected="selected" '; } **echo** '<option value="'.$value.'" '.$selected.'>'.$value.'</option>'; } ?> </select><br><br> | Selecting the currency (by default, it is the one choosed during subscription). |
|  <input type="hidden" name="MAX\_FILE\_SIZE" value="1048576" /> Upload payment proof if any : <input name="payment\_file" type="file"> <br> | Html input for the file upload. This is not mendatory as it is for manual payment. |
|  <div style="text-align:center; width:100%"> <input type="hidden" name="iws" value="<?= $wpbm->web\_subscription['i\_web\_subscription'] ?>" > <input type="hidden" name="action" value="create\_and\_pay" > <br><br><button class="pure-button" type="button" onclick="window.opener.location.reload(true);window.close()">Cancel</button> &nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp; &nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp; <input class="pure-button pure-button-primary" type="submit" value="Create and make payment"> </div> <?php $wpbm->\_close(); } ?> </form> | Validation of the form. |
|  <?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' ){ | If the action is to create the account and pay … |
|  $creation = **false**; **if** ( ! isset( $\_POST['amount'] ) || ! preg\_match('/^[0-9]+(\.[0-9]{0,2})?$/', $\_POST['amount'] ) ){ $ERROR\_MESSAGE = "NO AMOUNT DEFINED"; | We verify the amount. |
|  } **else** { $wpbm = **new** WebSubscriptions\_PortaBilling\_Management(); $wpbm->set\_i\_web\_subscription( $\_POST['iws'] ); $wpbm->get\_web\_subscription(); | We get the subscription from the database through the management class. |
|  **if** ( $wpbm->web\_subscription['Account\_Type'] != 'pre' ){  $ERROR\_MESSAGE = "This is not a prepaid account !"; | 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'],$web\_subscription\_file); | We upload the payment proof if any. |
|  **if** ($resultat){ $FILE\_UPLOAD\_ERROR = **FALSE**; $wpbm->update\_web\_subscription\_active\_file( $filename ); } } } | If upload is ok, we update the database for retrieving the document corresponding to this subscription latter. |
|  **if** ( $FILE\_UPLOAD\_ERROR == **FALSE** ){ $creation = $wpbm->create\_customer\_and\_account\_from\_web\_subscription(); | We create the customer. |
|  **if** ( $creation == **TRUE** ){  $wpbm->send\_creation\_notification( "prepaid" );  | Sending the notification to customer. |
|   $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); } **catch** (SoapFault $fault) { $ERROR\_MESSAGE = "Error during payment !\n\n".$fault; }  $ServiceCustomer->\_logout(); } } } $wpbm->\_close(); } | Processing the payment. |
|  ?> <?php **if** ( $ERROR\_MESSAGE == "" ){ ?> <h2>Account creation info</h2><br><br> Account has been created and payment has been added in billing.<br><br> <button class="pure-button" type="button" onclick="window.opener.location.reload(true);window.close()">Close</button> | Account creation and payment is done. |
|  <?php } **else** { ?> <h2>ERROR</h2><br><br> <?= $ERROR\_MESSAGE ?> <br><br> <button class="pure-button" type="button" onclick="window.opener.location.reload(true);window.close()">Close</button> | Displaying the error message if any. |
|  <?php }  } ?> </div></center></body></html> | End of script. |
|  |  |

## Page for opening account on signed contract reception

|  |  |
| --- | --- |
| CODE | COMMENT |
| <?php **if**(session\_id() == '') {session\_start();} | Session start  |
| <?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";?> | Includes. |
| <!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]> <script src="http://ie7-js.googlecode.com/svn/version/2.1(beta4)/IE7.js"></script> <![endif]--></head> | HTML header. |
| <body> <div id="head" style="padding:15px; width:80%; margin: 0 auto;" > <!-- <img src="/public/060801-web/images/switzernet.gif" style="width:100px" alt="Switzernet"> --> <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); ?> | Getting parameters from the url. |
|  <h2>Subscription info</h2><br><br> <?php $i = 1; **foreach** ($wpbm->web\_subscription['subscribedNumbers'] **as** $value) { **echo** 'Account '.$i.': '. $value['h323\_id'].'<br><br>'; $i++; } **echo** 'Firstname : '. $wpbm->web\_subscription['Firstname'].'<br><br>'; **echo** 'Lastname : '. $wpbm->web\_subscription['Lastname'].'<br><br>'; ?> <input type="hidden" name="MAX\_FILE\_SIZE" value="1048576" /> Upload signed contract : <input name="contract\_file" type="file" required> <br> <div style="text-align:center; width:100%"> <input type="hidden" name="iws" value="<?= $wpbm->web\_subscription['i\_web\_subscription'] ?>" > <input type="hidden" name="action" value="upload\_contract\_and\_create" > <br><br><button class="pure-button" type="button" onclick="window.opener.location.reload(true); window.close()">Cancel</button> &nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp; &nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp; <input class="pure-button pure-button-primary" type="submit" value="Create customer account"> </div> <?php $wpbm->\_close(); } ?> </form> | Displaying basic information about the subscription. |
|  <?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'] == 'upload\_contract\_and\_create' ){ | If action is to create. |
|  $creation = **false**; **if** ( ! isset( $\_FILES['contract\_file'] ) ){ $ERROR\_MESSAGE = "UPLOAD CONTRACT !"; | There is no contract uploaded. We display an error message. |
|  } **else** { $wpbm = **new** WebSubscriptions\_PortaBilling\_Management(); $wpbm->set\_i\_web\_subscription( $\_POST['iws'] ); $wpbm->get\_web\_subscription(); $FILE\_UPLOADED = **FALSE**; | Opening a web subscription management instance. |
|  **if** ( $wpbm->web\_subscription['Account\_Type'] == 'pre' ){  $ERROR\_MESSAGE = "This is a prepaid account !"; | The account must not be prepaid. |
|  } **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'],$web\_subscription\_file); | Uploading the contract and saving. |
|  **if** ($resultat){ $FILE\_UPLOADED = **TRUE**; $wpbm->update\_web\_subscription\_active\_file( $filename ); } | If the upload succeed we update the database. |
|  } **if** ( $FILE\_UPLOADED ){ $creation = $wpbm->create\_customer\_and\_account\_from\_web\_subscription();  | Then we create the customer in the billing. |
|  **if** ( $creation == **TRUE** ){  $wpbm->send\_creation\_notification( "contract" );  } | And finally send the account creation notification. |
|  } } $wpbm->\_close(); } ?> | Closing the web subscription instance. |
|  <?php **if** ( $ERROR\_MESSAGE == "" ){ ?> <h2>Account creation info</h2><br><br> Account has been created. Please verify in porta-billing.<br><br> | Informing the account has been created. |
|  <button class="pure-button" type="button" onclick="window.opener.location.reload(true);window.close()">Close</button> | Button for closing the windows. |
|  <?php } **else** { ?> <h2>ERROR</h2><br><br> <?= $ERROR\_MESSAGE ?> <br><br> | Displaying the error is the account creation failed. |
|  <button class="pure-button" type="button" onclick="window.opener.location.reload(true);window.close()">Close</button> <?php }  } ?> </div> | Button for closing the windows. |
| </center></body></html> | End of the page. |

# Liens

Ce document :

[http://switzernet.com/3/public/141015-automatic-web-subscriptions/](http://switzernet.com/3/public/folder/)

\* \* \*



Copyright © 2014 by Switzernet