

MailBoxValidator™ Email Validation Web Service

1. Overview	3
1.1 Overview of the MailBoxValidator Web Service	4
1.2 Front End of MailBoxValidator Email Validation Web Service	4
1.2.1 Integration with In-house System	4
1.2.2 Direct Access to Hosted Web Service	6
1.3 Back End of MailBoxValidator Email Validation Web Service	7
1.4 Process Flow Overview	7
2. Implementation	8
2.1 Implementation Details	9
2.2 Basic Parameters - Input	11
2.3 Basic Parameters – Output	11
2.4 List of possible value for MESSAGE field	11
2.5 Structure of the Request and Response	11
2.5.1 Request by SOAP	11
2.5.2 Response by SOAP	12
2.5.3 Request by HTTP-GET	12
2.5.4 Response by HTTP-GET	12
2.5.5 Request by HTTP-POST	13
2.5.6 Response by HTTP-POST	13
3. Design Information	14
3.1 Placement of MailBoxValidator Email Validation Web Service	14
Appendix I : Sample Code	15

1. Overview

This documentation provides a basic understanding and information to help you get started with our products. Look over this documentation to gain a high-level understanding of the process flow that underlies the MailBoxValidator Email Validation Web Service.

For more information, please visit <http://www.fraudlabs.com> or contact your MailBoxValidator representative:

Email : sales@fraudlabs.com

1.1 Overview of the MailBoxValidator Web Service

MailBoxValidator delivers scalable web services solutions that help you check if an email address is a valid one. It finds invalid email addresses without actually sending an email and make sure you keep valid email addresses of your customers.

MailBoxValidator Email Validation Web Service is hosted, programmable XML Web Service that allows exchange of data between systems. It is hosted on redundant servers and 24x7x365 monitoring. MailBoxValidator is highly scalable email validation solution and it can be easily integrated into your website to check addresses in the real time.

Key Features Include:

- Validates the incorrect email address formats and syntax
- Validates DNS domain by checking MX records
- SMTP checking
- Provides a complete XML-based Web Services API
- Contains sample code examples for ease integration

1.2 Front End of MailBoxValidator Email Validation Web Service

The general idea is that front end is responsible for collection input from the user and conforms to some specification that the back end can use. Front end of MailBoxValidator Email Validation Web Service is rather simple to understand. As we are using platform independent XML format to exchange data between systems, you may either integrate our web service to your in-house system, or direct access to our hosted web service.

1.2.1. Integration with In-house System

- I. Our sample codes in different languages are available at: http://www.fraudlabs.com/mailboxvalidator_samplecodes.aspx . Log on and download sample codes that you need (For sample codes in different languages please refer to Appendix II). Below are links to get sample codes:

- i. Microsoft ASP.NET - VB.NET:
<http://www.fraudlabs.com/samplecode/MailBoxValidatorWebServiceClientVB.zip>
 - ii. Microsoft ASP.Net - C#:
<http://www.fraudlabs.com/samplecode/MailBoxValidatorWebServiceClientCSharp.zip>
 - iii. Microsoft ASP 3.0:
<http://www.fraudlabs.com/samplecode/MailBoxValidatorWebServiceClientAsp.zip>
 - iv. Java/ Apache:
<http://www.fraudlabs.com/samplecode/MailBoxValidatorWebServiceClientJava.zip>
 - v. PHP:
<http://www.fraudlabs.com/samplecode/MailBoxValidatorWebServiceClientPHP.zip>
 - vi. Python:
<http://www.fraudlabs.com/samplecode/MailBoxValidatorWebServiceClientPython.zip>
 - vii. Perl:
<http://www.fraudlabs.com/samplecode/MailBoxValidatorWebServiceClientPerl.zip>
 - viii. ColdFusion MX:
<http://www.fraudlabs.com/samplecode/MailBoxValidatorWebServiceClientColdFusion.zip>
 - ix. Access 2000:
<http://www.fraudlabs.com/samplecode/MailBoxValidatorWebServiceClientAccess.zip>
 - x. Excel 2000:
<http://www.fraudlabs.com/samplecode/MailBoxValidatorWebServiceClientExcel.zip>
- II. Please go through 'readme' file that we provide together with the sample codes for more set up information
- III. In order to use our service, you need to get your own license key – *How?*
- i. log on to <http://www.fraudlabs.com>
 - ii. sign up as our registered member
 - iii. check your email to complete user activation
 - iv. get license key :
 - a) Free License Account:
 - 1) at the left menu bar, under category of MailBoxValidator™ Email Validation, click "free license"

- 2) view Terms of Use (*please note that you must agree with our Terms of Use before proceed)
 - 3) click on "Get Free License Now"
 - 4) the license key will be sent to your email
- b) Premium Subscription:
- 1) at the left menu bar, under category of MailBoxValidator™ Mexico, click "subscribe now"
 - 2) fill in required field in the secure payment form
 - 3) click on "make payment"
 - 4) the license key will be sent to your email after your payment is confirmed by our online payment merchant

IV. Now fill in the required field

V. Click on "Submit" and result is provided

1.2.2. Direct Access to Hosted Web Service

- I. To use our web service directly, please log on to:
<http://ws2.fraudlabs.com/MailBoxValidator.aspx>
- II. Click on "MailBoxValidator"
- III. In order to use our service, you need to get your own license key – *How?*
 - i. log on to <http://www.fraudlabs.com>
 - ii. sign up as our registered member
 - iii. check your email to complete user activation
 - iv. get license key :
 - a) Free License Account:
 - 1) at the left menu bar, under category of MailBoxValidator™ Email Validation, click "free license"
 - 2) view Terms of Use (*please note that you must agree with our Terms of Use before proceed)
 - 3) click on "Get Free License Now"
 - 4) the license key will be sent to your email
 - b) Premium Subscription:
 - 1) at the left menu bar, under category of MailBoxValidator™ Email Validation, click "subscribe now"
 - 2) fill in required field in the secure payment form
 - 3) click on "make payment"

- 4) the license key will be sent to your email after your payment is confirmed by our online payment merchant

IV. Now you fill in the particular field

V. Click on “Invoke” and result is provided

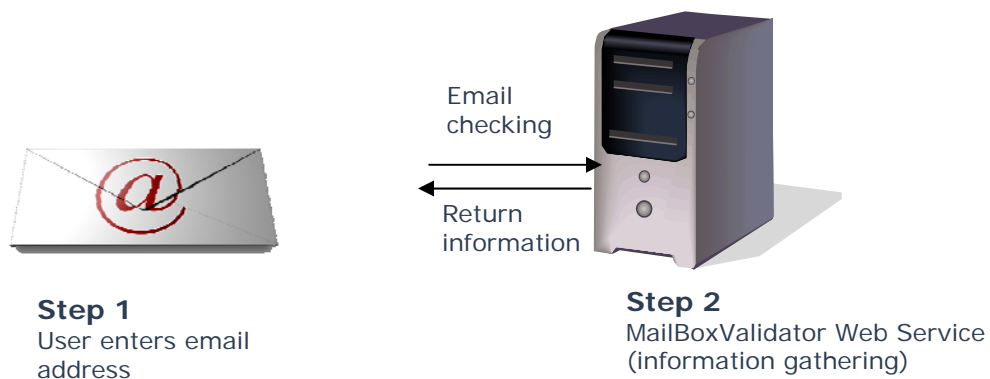
1.3 Back End of MailBoxValidator Email Validation Web Service

Back end is the part that processes the input from the front end. The process is seamless to the end-user and works by interaction with a SOAP API through MailBoxValidator Email Validation Web Service.

The process works as follows:

1. User enters email that to be validate
2. System verify the license key before proceed
3. If the license key is valid, it will proceed and check the credits availability. If the remaining credits still available, it will start to retrieve related information from database
4. Returns and display the precise information

1.4 Process Flow Overview



2. Implementation

This section provides basic information of the process of integrating the web service into your website. Look over this section to gain a high-level understanding of requesting MailBoxValidator Email Validation Web Service.

The implementation section covers the following topics:

- MailBoxValidator Function

For those with solid SOAP programming knowledge, integration basics are included below. POST all requests to:

<http://ws2.fraudlabs.com/MailBoxValidator.asmx>

For more information about MailBoxValidator Email Validation Web Service implementation, please visit <http://www.fraudlabs.com> or contact your MailBoxValidator representative:

Email : sales@fraudlabs.com

2.1 Implementation Details

This section was created for those who wish to develop applications that make use of MailBoxValidator Email Validation Web Service.

SOAP – MailBoxValidator Function Request

```
POST /mailboxvalidator.asmx HTTP/1.1
Host: ws2.fraudlabs.com
Content-Type: text/xml; charset=utf-8
Content-Length: length
SOAPAction: "http://ws.fraudlabs.com/MailBoxValidator"

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <MailBoxValidator xmlns="http://ws.fraudlabs.com/">
      <EMAIL>string</EMAIL>
      <LICENSE>string</LICENSE>
    </MailBoxValidator>
  </soap:Body>
</soap:Envelope>
```

SOAP – MailBoxValidator Function Response

```
HTTP/1.1 200 OK
Content-Type: text/xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <MailBoxValidatorResponse xmlns="http://ws.fraudlabs.com/">
      <MailBoxValidatorResult>
        <IS_SYNTAX>string</IS_SYNTAX>
        <IS_DOMAIN>string</IS_DOMAIN>
        <IS_SMTP>string</IS_SMTP>
        <IS_LEVEL>string</IS_LEVEL>
        <CREDITSAVAILABLE>string</CREDITSAVAILABLE>
        <MESSAGE>string</MESSAGE>
      </MailBoxValidatorResult>
    </MailBoxValidatorResponse>
  </soap:Body>
</soap:Envelope>
```

HTTP GET- MailBoxValidator Function Request

```
GET /mailboxvalidator.asmx/MailBoxValidator?EMAIL=string&LICENSE=string
HTTP/1.1
Host: ws2.fraudlabs.com
```

HTTP GET- MailBoxValidator Function Response

```
HTTP/1.1 200 OK
Content-Type: text/xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<MBVALIDATOR xmlns="http://ws.fraudlabs.com/">
  <IS_SYNTAX>string</IS_SYNTAX>
  <IS_DOMAIN>string</IS_DOMAIN>
  <IS_SMTP>string</IS_SMTP>
  <IS_LEVEL>string</IS_LEVEL>
  <CREDITSAVAILABLE>string</CREDITSAVAILABLE>
  <MESSAGE>string</MESSAGE>
</MBVALIDATOR>
```

HTTP POST- MailBoxValidator Function Request

```
POST /mailboxvalidator.asmx/MailBoxValidator HTTP/1.1
Host: ws2.fraudlabs.com
Content-Type: application/x-www-form-urlencoded
Content-Length: length

EMAIL=string&LICENSE=string
```

HTTP POST- MailBoxValidator Function Response

```
HTTP/1.1 200 OK
Content-Type: text/xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<MBVALIDATOR xmlns="http://ws.fraudlabs.com/">
  <IS_SYNTAX>string</IS_SYNTAX>
  <IS_DOMAIN>string</IS_DOMAIN>
  <IS_SMTP>string</IS_SMTP>
  <IS_LEVEL>string</IS_LEVEL>
  <CREDITSAVAILABLE>string</CREDITSAVAILABLE>
  <MESSAGE>string</MESSAGE>
</MBVALIDATOR>
```

A WSDL is available at:

<http://ws2.fraudlabs.com/mailboxvalidator.asmx?wsdl>

A description of the MailBoxValidator operation is available at:
<http://ws2.fraudlabs.com/mailboxvalidator.asmx?op=MailBoxValidator>

2.2 Basic Parameters - Input

Field	Format	Description
EMAIL	Required	Email address that to be verified.
LICENSE	Required	License key for free license and premium users.

2.3 Basic Parameters – Output

Field	Format	Description
IS_SYNTAX	string	Return True or False to determine if the syntax is correct
IS_DOMAIN	string	Return True or False to determine if the domain name has MX records
IS_SMTP	string	Return True or False to determine if the SMTP server is working
IS_LEVEL	string	Return True or False to determine if the email is bad address
CREDITSAVAILABLE	integer	Number of queries remaining in your account, can be used to alert you when you may need to add more queries to your account.
MESSAGE	string	Web Service Message Response

2.4 List of possible value for MESSAGE field

MailBoxValidator Web Service Error Message
• Error: Domain Checking Failed
• Error: Syntax checking failed
• Error: Domain checking failed, connection attempt failed
• No credit available
• License key was expired

2.5 Structure of the Request and Response

2.5.1. Request by SOAP

A valid request posted from user should look something like this:

```
<?xml version="1.0" encoding="utf-8"?>
```

```
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
  <soap12:Body>
    <MailBoxValidator xmlns="http://ws.fraudlabs.com/">
      <EMAIL>sales@fraudlabs.com</EMAIL>
      <LICENSE>xx-xxxx-xxxx</LICENSE>
    </MailBoxValidator>
  </soap12:Body>
</soap12:Envelope>
```

2.5.2. Response by SOAP

After posting a valid request, MailBoxValidator will return several parameters. An example of a response returned MailBoxValidator is shown below:

```
<?xml version="1.0" encoding="utf-8"?>
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
  <soap12:Body>
    <MailBoxValidatorResponse xmlns="http://ws.fraudlabs.com/">
      <MailBoxValidatorResult>
        <IS_SYNTAX>True</IS_SYNTAX>
        <IS_DOMAIN>True</IS_DOMAIN>
        <IS_SMTP>True</IS_SMTP>
        <IS_LEVEL>3</IS_LEVEL>
        <CREDITSAVAILABLE>123</CREDITSAVAILABLE>
        <MESSAGE/>
      </MailBoxValidatorResult>
    </MailBoxValidatorResponse>
  </soap12:Body>
</soap12:Envelope>
```

2.5.3. Request by HTTP-GET

A valid request posted by user:

```
GET
/ws2.fraudlabs.com_non_ssl/MailBoxValidator.asmx/MailBoxValidator?Email
=sales@fraudlabs.com&LICENSE=xx-xxxx-xxxx HTTP/1.1
Host: ws2.fraudlabs.com
```

2.5.4. Response by HTTP-GET

An example of a response returned MailBoxValidator_Mexico is shown below:

```
HTTP/1.1 200 OK
Content-Type: text/xml; charset=utf-8
Content-Length: length
```

```
<?xml version="1.0" encoding="utf-8"?>
<MBVALIDATOR xmlns="http://ws.fraudlabs.com/">
  <IS_SYNTAX>True</IS_SYNTAX>
  <IS_DOMAIN>True</IS_DOMAIN>
  <IS_SMTP>True</IS_SMTP>
  <IS_LEVEL>3</IS_LEVEL>
  <CREDITSAVAILABLE>123</CREDITSAVAILABLE>
  <MESSAGE />
</MBValidator >
```

2.5.5. Request by HTTP-POST

A valid request posted by user:

```
POST /ws2.fraudlabs.com_non_ssl/MailBoxValidator.asmx/MailBoxValidator
HTTP/1.1
Host: ws2.fraudlabs.com
Content-Type: application/x-www-form-urlencoded
Content-Length: length

Email=sales@fraudlabs.com&LICENSE=XX-XXXX-XXXX
```

2.5.6. Response by HTTP-POST

An example of a response returned by MailBoxValidator_Mexico is shown below:

```
HTTP/1.1 200 OK
Content-Type: text/xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<MBVALIDATOR xmlns="http://ws.fraudlabs.com/">
  <IS_SYNTAX>True</IS_SYNTAX>
  <IS_DOMAIN>True</IS_DOMAIN>
  <IS_SMTP>True</IS_SMTP>
  <IS_LEVEL>3</IS_LEVEL>
  <CREDITSAVAILABLE>123</CREDITSAVAILABLE>
  <MESSAGE/>
</MBVALIDATOR>
```

3. Design Information

This section provides suggestion regarding the placing of MailBoxValidator Email Validation Web Service into Web pages. Look over this section to get a general idea for adding this service to your website.

3.1 Placement of MailBoxValidator Email Validation Web Service

As MailBoxValidator Email Validation Web Service provides you very informative details by using Email, this service can be placed on any forms on your website, offering a seamless integration.

Level of security: High

Site Type(s):

- e-Commerce Solutions
- Internet Retailers
- Software Developers

Benefits:

- Get rid of undelivered mails
- Validate correct format / syntax of emails
- Validate domain for DNS and MX lookups
- Validate SMTP if it accepts connection
- Improve data entry speed and accuracy

Appendix I : Sample Code

MailBoxValidator Web Service sample code is available in several different programming languages. Below are some examples, for more different programming languages please log on to:

<http://www.fraudlabs.com/mailboxvalidatorsamplecodes.aspx>

i. ASP.NET – VB.NET (SOAP)

```
Private Sub MailBoxValidator()  
    Dim x_MailBoxValidator As New MailBoxValidator. MailBoxValidator  
    Dim o_MailBoxValidator As New MailBoxValidator  
  
    Try  
        oMailBoxValidator = x_MailBoxValidator.MailBoxValidator  
        (Me.txtEmail.Text, Me.txtLicense.Text)  
  
        Me.txtResult.Text = "IS_SYNTAX:" + oMailBoxValidator.IS_SYNTAX &  
        vbNewLine  
        Me.txtResult.Text += "IS_DOMAIN:" + oMailBoxValidator.IS_DOMAIN &  
        vbNewLine  
        Me.txtResult.Text += "IS_SMTP:" + oMailBoxValidator.IS_SMTP & vbNewLine  
        Me.txtResult.Text += "IS_LEVEL:" + oMailBoxValidator.IS_LEVEL &  
        vbNewLine  
        Me.txtResult.Text += "Credits Available:" +  
        oMailBoxValidator.CREDITSAVAILABLE & vbNewLine  
        Me.txtResult.Text += "Message:" + oMailBoxValidator.MESSAGE & vbNewLine  
  
        Catch ex As Exception  
            Response.Write(ex.Message)  
        End Try  
    End Sub
```

ii. ASP.NET – VB.NET (HTTP)

```
Private Sub Lookup()  
    Dim x_MailBoxValidator As New MailBoxValidator.MailBoxValidator  
    Dim oMailBoxValidator As New MailBoxValidator  
  
    Try  
        ' create the request URL  
        Dim x_builder As String  
        ' add the host element of the URL  
        x_builder =  
        "http://ws2.fraudlabs.com/MailBoxValidator.asmx/MailBoxValidator?" & _  
        "EMAIL=" + Me.txtEmail.Text & _  
        "&LICENSE=" + Me.txtLicense.Text  
        ' create the web client and obtain the response data  
        ' as a byte array  
        Dim x_web_client As WebClient = New WebClient  
        Dim x_response() As Byte =  
        x_web_client.DownloadData(x_builder.ToString())  
        ' process the string result to obtain a validation result
```

```
processResultString(Encoding.Default.GetString(x_response))
Catch ex As Exception
    Response.Write(ex.Message)
End Try
End Sub

Private Sub processResultString(ByVal p_result As String)
Dim indexOpen As Integer
Dim indexClose As Integer
Dim strParam As String

strParam = "<IS_SYNTAX>"
indexOpen = p_result.IndexOf(strParam) + strParam.Length
indexClose = p_result.IndexOf("</IS_SYNTAX>")
Me.txtResult.Text = "IS_SYNTAX: "
If indexClose <> -1 Then
Me.txtResult.Text += p_result.Substring(indexOpen, indexClose -
indexOpen)
End If
strParam = "<IS_DOMAIN>"
indexOpen = p_result.IndexOf(strParam) + strParam.Length
indexClose = p_result.IndexOf("</IS_DOMAIN>")
Me.txtResult.Text += vbNewLine & "IS_DOMAIN: "
If indexClose <> -1 Then
Me.txtResult.Text += p_result.Substring(indexOpen, indexClose -
indexOpen)
End If
strParam = "<IS_SMTTP>"
indexOpen = p_result.IndexOf(strParam) + strParam.Length
indexClose = p_result.IndexOf("</IS_SMTTP>")
Me.txtResult.Text += vbNewLine & "IS_SMTTP: "
If indexClose <> -1 Then
Me.txtResult.Text += p_result.Substring(indexOpen, indexClose -
indexOpen)
End If
strParam = "<IS_LEVEL>"
indexOpen = p_result.IndexOf(strParam) + strParam.Length
indexClose = p_result.IndexOf("</IS_LEVEL>")
Me.txtResult.Text += vbNewLine & "IS_LEVEL: "
If indexClose <> -1 Then
Me.txtResult.Text += p_result.Substring(indexOpen, indexClose -
indexOpen)
End If
strParam = "<CREDITSAVAILABLE>"
indexOpen = p_result.IndexOf(strParam) + strParam.Length
indexClose = p_result.IndexOf("</CREDITSAVAILABLE>")
Me.txtResult.Text += vbNewLine & "Credits Available:"
If indexClose <> -1 Then
Me.txtResult.Text += p_result.Substring(indexOpen, indexClose -
indexOpen)
End If
strParam = "<MESSAGE>"
indexOpen = p_result.IndexOf(strParam) + strParam.Length
indexClose = p_result.IndexOf("</MESSAGE>")
Me.txtResult.Text += vbNewLine & "Message: "
If indexClose <> -1 Then
Me.txtResult.Text += p_result.Substring(indexOpen, indexClose -
```

```
indexOpen)
End If
End Sub
```

iii. ASP.NET – C#.NET (SOAP)

```
private void MailBoxValidatorWebService()
{
MailBoxValidator.MailBoxValidator x_MailBoxValidator = new
MailBoxValidator.MailBoxValidator();

MailBoxValidator oMailBoxValidator = new MailBoxValidator();

try
{
oMailBoxValidator = x
oMailBoxValidator.oMailBoxValidator(this.txtEmail.Text,
this.txtLicense.Text);

this.txtResult.Text = "IS_SYNTAX:" + oMailBoxValidator.IS_SYNTAX +
"\n";
this.txtResult.Text += "IS_DOMAIN:" + oMailBoxValidator.IS_DOMAIN +
"\n";
this.txtResult.Text += "IS_SMTP:" + oMailBoxValidator.IS_SMTP + "\n";
this.txtResult.Text += "IS_LEVEL:" + oMailBoxValidator.IS_LEVEL + "\n";
this.txtResult.Text += "Credits Available:" +
oMailBoxValidator.CREDITSAVAILABLE + "\n";
this.txtResult.Text += "Message:" + oMailBoxValidator.MESSAGE + "\n";
}
catch (Exception ex)
{
Response.Write(ex.Message);
}
}
```

iv. ASP.NET – C#.NET (HTTP)

```
private void Lookup()
{
MailBoxValidator.MailBoxValidator x_MailBoxValidator = new
MailBoxValidator.MailBoxValidator();

MailBoxValidator oMailBoxValidator = new MailBoxValidator();

try
{
//create the request URL
string x_builder;
//add the host element of the URL
x_builder =
"http://ws2.fraudlabs.com/MailBoxValidator.asmx/MailBoxValidator?Postal
CODE=" + this.txtEmail.Text + "&LICENSE=" + this.txtLicense.Text;

// create the web client and obtain the response data as a byte array
```

```
WebClient x_web_client = new WebClient();
byte [] x_response = x_web_client.DownloadData(x_builder.ToString());
// process the string result to obtain a validation result

processResultString(Encoding.Default.GetString(x_response));
}
catch (Exception ex)
{
    Response.Write(ex.Message);
}
}

private void processResultString(String p_result)
{
    int indexOpen;
    int indexClose;
    string strParam;
    strParam = "<IS_SYNTAX>";
    indexOpen = p_result.IndexOf(strParam) + strParam.Length;
    indexClose = p_result.IndexOf("</IS_SYNTAX>");
    this.txtResult.Text = "IS_SYNTAX: ";
    if (indexClose != -1)
    {
        this.txtResult.Text += p_result.Substring(indexOpen, indexClose -
indexOpen);
    }
    strParam = "<IS_DOMAIN>";
    indexOpen = p_result.IndexOf(strParam) + strParam.Length;
    indexClose = p_result.IndexOf("</IS_DOMAIN>");
    this.txtResult.Text += "\rIS_DOMAIN: ";
    if (indexClose != -1)
    {
        this.txtResult.Text += p_result.Substring(indexOpen, indexClose -
indexOpen);
    }
    strParam = "<IS_SMTP>";
    indexOpen = p_result.IndexOf(strParam) + strParam.Length;
    indexClose = p_result.IndexOf("</IS_SMTP>");
    this.txtResult.Text += "\rIS_SMTP: ";
    if (indexClose != -1)
    {
        this.txtResult.Text += p_result.Substring(indexOpen, indexClose -
indexOpen);
    }
    strParam = "<IS_LEVEL>";
    indexOpen = p_result.IndexOf(strParam) + strParam.Length;
    indexClose = p_result.IndexOf("</IS_LEVEL>");
    this.txtResult.Text += "\rIS_LEVEL: ";
    if (indexClose != -1)
    {
        this.txtResult.Text += p_result.Substring(indexOpen, indexClose -
indexOpen);
    }
    strParam = "<CREDITSAVAILABLE>";
    indexOpen = p_result.IndexOf(strParam) + strParam.Length;
    indexClose = p_result.IndexOf("</CREDITSAVAILABLE>");
    this.txtResult.Text += "\rCredits Available:";
```

```
if (indexClose != -1)
{
this.txtResult.Text += p_result.Substring(indexOpen, indexClose -
indexOpen);
}
strParam = "<MESSAGE>";
indexOpen = p_result.IndexOf(strParam) + strParam.Length;
indexClose = p_result.IndexOf("</MESSAGE>");
this.txtResult.Text += "\rMessage: ";
if (indexClose != -1)
{
this.txtResult.Text += p_result.Substring(indexOpen, indexClose -
indexOpen);
}
}
```

v. PHP

```
<?php
if (!isset($_POST['submit'])) {} // if page is not submitted to itself
echo the form
else
{
$Email = $_POST["Email"];
$License = $_POST["License"];

if ($License == "<Enter License Key>" || $License == "")
{
    echo "License key are required field." ;
}

else
{
require_once('./lib/nusoap.php');
// Call the service with this message
$msg = '<MailBoxValidator xmlns="http://ws2.fraudlabs.com/">
<EMAIL>'.$Email.'</EMAIL>
<LICENSE>'.$License.'</LICENSE>
</MailBoxValidator>';

// Create a new SOAP object
$soapclient = new
soapclient_nusoap('http://ws2.fraudlabs.com/MailBoxValidator.asmx?wsdl'
, 'wsdl');
if (!$soapclient->fault) {
$result = $soapclient->call('MailBoxValidator, array($msg));
if(!isset($result['Error'])) {
// No Error. Retrieve results.
echo "IS_SYNTAX = " . $result["IS_SYNTAX"] . "<br>";
echo "IS_DOMAIN = " . $result["IS_DOMAIN"] . "<br>";
echo "IS_SMTP = " . $result["IS_SMTP"] . "<br>";
echo "IS_LEVEL = " . $result["IS_LEVEL"] . "<br>";
echo "CREDITSAVAILABLE = " . $result["CREDITSAVAILABLE"] . "<br>";
echo "MESSAGE = " . $result["MESSAGE"] . "<br>";
}
}
```

```
else {
// Error
echo "Error Description = " . $result["Error"]["Desc"] . "<br>";
echo "Error Number = " . $result["Error"]["Number"] . "<br>";
}
}
else {
echo "Error = {$soapclient->faultcode}<br>";
echo "String = {$soapclient->faultstring}";
} // end if $soapclient->call is success
}
}
?>
```

vi. JAVA / APACHE

```
import java.io.*;

//Web Service Client Class
public class Client
{
//Entry Point to this Application
public static void main(String[] args)
{
try
{
//Create a proxy
ApacheSoapProxy MBValidator = new ApacheSoapProxy ();

//Invoke FraudLabsCheck() over SOAP and get the new OID
//Require Parameter FraudLabsCheck(String Email, String LICENSE)
String result = MBValidator.MailBoxValidator("test@test.com",
"<Enter License Key>");

//Print out the value
System.out.println (result);
}
catch (java.net.MalformedURLException exception)
{
exception.printStackTrace ();
}
catch (org.apache.soap.SOAPException exception)
{
exception.printStackTrace ();
}
}
}
```