

# **IP2Currency™ Exchange Rate Web Service**

~~ Exchange Rate Made Easy! ~~

<b>1. Overview</b>	<b>3</b>
1.1 Overview of the IP2Currency Web Service	4
1.2 Front End of IP2Currency Web Service	5
Integration with In-house System	5
1.3 Back End of IP2Currency Web Service	7
1.3.1. Get the exchange rate by visitor's IP address	7
1.3.2. Get the converted value by visitor's IP address	7
1.3.3. Get custom exchange rate	8
1.3.4. Get custom converted value	9
1.3.5. Get system defined currency info by visitor's IP address	9
1.4 Process Flow Overview	9
<b>2. Implementation</b>	<b>11</b>
2.1 Basic Parameters - Input	12
2.2 Basic Parameters - Output	12
2.3 List of possible value for MESSAGE field	13
<b>3. Design Information</b>	<b>14</b>
3.1 Placement of IP2Currency Web Service	14
<b>Appendix I: ISO3166 Country Code</b>	<b>15</b>
<b>Appendix II: ISO4127 Currency Code</b>	<b>21</b>
<b>Appendix III: Sample Code</b>	<b>23</b>

## **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 IP2Currency Web Services.

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

Email : [sales@fraudlabs.com](mailto:sales@fraudlabs.com)

## **1.1 Overview of the IP2Currency Web Service**

IP2Currency is a hosted XML-Based Web API providing user an easy way to get the localized & latest Currency Exchange Rate based on visitor's IP address. There is no need for you to perform an additional step to find the visitor's country, as it will be automatically performed by our server which will determine the country the visitor originates from, and what is the commonly used currency, and the currency symbol.

The use of XML-Based Web Service enables you to easily integrate our solution regardless of any web server and other business solution platform. It is very easy to set up and use.

This solution will be very useful and a *must-have* if you are going to set up an online web store selling products globally and accepting multiple currencies. Imagine that what will you feel if you are able to view and browse the products using your local currency? Obviously people will like it as they know exactly how much they are going to pay in their preferred currency.

Furthermore, our Web Service provides you an accurate daily exchange rate that saves you a lot of efforts to source for the reliable data providers.

### **Key Features Include:**

- Get exchange rate by visitor's IP address
- Get converted amount by visitor's IP address
- Get custom exchange rate
- Get custom converted amount
- Get system defined currency info by visitor's IP address

## 1.2 Front End of IP2Currency Web Service

The general idea is that the front end is responsible for collecting input from the user and conforms to some specification that the back end can use. The front end of IP2Currency 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.

### Integration with In-house System

- I. Our sample codes in different languages are available at: <http://www.fraudlabs.com/ip2currencysamplecodes.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 (v1.1) - VB.NET:  
<http://www.fraudlabs.com/samplecode/ip2currencywebserverviceclientvb.zip>
  - ii. Microsoft ASP.Net (v1.1) - C#:  
<http://www.fraudlabs.com/samplecode/ip2currencywebserverviceclientcsharp.zip>
  - iii. Microsoft ASP.NET (v2.0) - VB.NET:  
<http://www.fraudlabs.com/samplecode/ip2currencywebserverviceclientvb2005.zip>
  - iv. Microsoft ASP.Net (2.0) - C#:  
<http://www.fraudlabs.com/samplecode/ip2currencywebserverviceclientcsharp2005.zip>
  - v. Java:  
<http://www.fraudlabs.com/samplecode/ip2currencywebserverviceclientjava.zip>
  - vi. PHP:  
<http://www.fraudlabs.com/samplecode/ip2currencywebserverviceclientphp.zip>
  - vii. Python:  
<http://www.fraudlabs.com/samplecode/ip2currencywebserverviceclientpython.zip>
  - viii. Perl:  
<http://www.fraudlabs.com/samplecode/ip2currencywebserverviceclientperl.zip>
  - ix. ColdFusion 9:

<http://www.fraudlabs.com/samplecode/ip2currencywebserverclientcoldfusion.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 IP2Currency™ Exchange Rate, 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 IP2Currency™ Exchange Rate, 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

### 1.3 Back End of IP2Currency 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 to access the IP2Currency Web Service.

The process works as follows:

1. User submits relevant input values and license key
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 process your web service request.
4. Returns the currency values

#### 1.3.1. Get the exchange rate by visitor's IP address

This feature will get the exchange rate by visitor's IP address. There is no need for you to supply the output currency, and it will be determined by our server of what are the commonly used output currency and the latest conversion rate. A generic solution to take only the visitor's IP address and your base currency, and magically returns you the result.

Below is the concept flow of how the currency rate is being determined behind the scene.



The input parameter needed

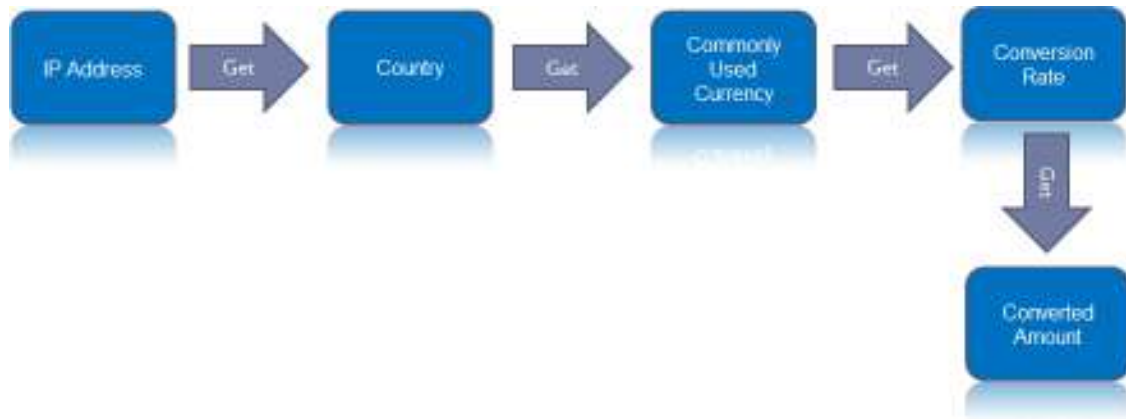
- LICENSE
- VISITORIP
- FROMCURRENCYCODE

#### 1.3.2. Get the converted value by visitor's IP address

This feature will get the converted value by visitor's IP address. It uses the same logic as the above feature (*Get the exchange rate by visitor's*

IP address), but added one additional function to convert the given amount.

Below is the concept flow

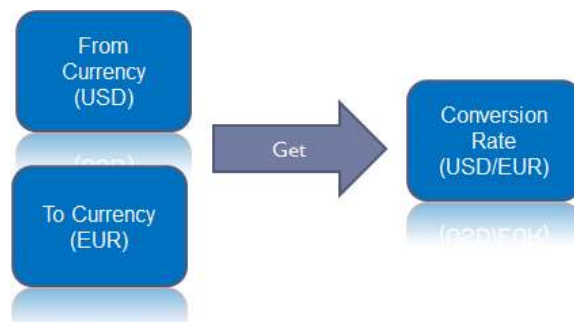


The input parameter needed

- LICENSE
- VISITORIP
- FROMCURRENCYCODE
- FROMAMOUNT

### 1.3.3. Get custom exchange rate

This feature allows you to explicitly query the currency pair exchange rate. For example, in some occasions you may want to know the exchange rate of a particular currency pair instead of the auto-detection feature described above.



The input parameter needed

- LICENSE
- FROMCURRENCYCODE
- TOCURRENCYCODE



### 1.3.4. Get custom converted value

This feature allows you to explicitly get the converted value from the currency pair supplied. For example, you want to know the converted amount from USD to EUR.



The input parameters needed

- LICENSE
- FROMCURRENCYCODE
- TOCURRENCYCODE
- AMOUNT

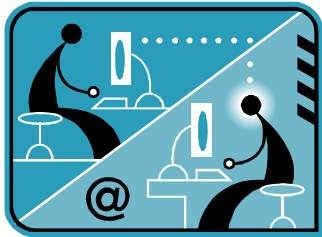
### 1.3.5. Get system defined currency info by visitor's IP address

The feature allows you to query the country name and the commonly used currency for a given IP address.

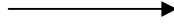
The input parameters needed

- LICENSE
- VISITORIP

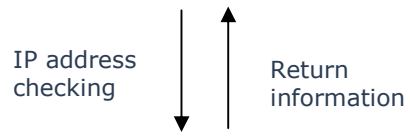
## 1.4 Process Flow Overview



**Step 1**  
Visitors visit your  
websites



**Step 2**  
Visitor's IP address  
captured by merchant



**Step 3**  
IP2Currency Web Service

## **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 IP2Currency Web Service.

A WSDL is available at:

<http://v1.fraudlabs.com/ip2currencywebservice.asmx?wsdl>

For more information about IP2Currency Web Service implementation, please visit <http://www.fraudlabs.com> or contact your FraudLabs representative:

Email: [sales@fraudlabs.com](mailto:sales@fraudlabs.com)

## 2.1 Basic Parameters - Input

Field	Format	Description
LICENSE	Required	License key for free license and premium users.
VISITORIP	Optional	Visitor IP address
FROMCURRENCYCODE	Optional	Base currency code (ISO 4217 standard)
TOCURRENCYCODE	Optional	Target currency code (ISO 4217 standard)
FROMAMOUNT	Optional	Currency amount for conversion

## 2.2 Basic Parameters - Output

Field	Format	Description
FROMCURRENCYCODE	Char(3)	Base currency code (ISO 4217 standard)
TOCURRENCYSYMBOL	String	Target currency symbol
TOCURRENCYCODE	Char(3)	Target currency code (ISO 4217 standard)
FROMAMOUNT	String	Amount before currency conversion
TOAMOUNT	Float	Amount after currency conversion
CONVERSIONRATE	Float	Currency exchange rate
TOCOUNTRYNAME	String	Full country name obtained via the IP address supplied
TOCOUNTRYCODE	Char(2)	2 character country code (ISO 3166 standard)
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

### 2.3 List of possible value for MESSAGE field

<b>Ip2Currency Web Service Error Message</b>
• License Key cannot be blank.
• Missing parameters.
• Invalid license key.
• Invalid IP address.
• Invalid currency code to convert from.
• Invalid currency code to convert to.
• Unable to find conversion rate.
• Invalid amount for conversion.
• The amount to convert is too big.
• No credit available
• License key has expired

### **3. Design Information**

This section provides suggestion regarding the placing of IP2Currency 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 IP2Currency Web Service**

IP2Currency web service provides you an easy way to get the localized & latest Currency Exchange Rate based on visitor's IP address.

**Description:** Visitors geographical information is needed to determine where the visitors are from. By having this information, the system can supply the localized currency information and the latest exchange rate to support your multi-currencies web solution.

**Level of security:** High

**Site Type(s):**

- e-Commerce Solution Providers
- Internet Retailers
- Software Developers

**Benefits:**

- Easy to Use
  - Single XML-Based web service Interface provides you a robust solution
- Easily Integrated
  - XML-Based Web Service to be easily integrated into most of the business solutions.
- Rate Accuracy
  - Our conversion rates derived from multiple sources ensuring you the accuracy of the exchange rate.
- Service Reliability
  - Our servers are hosted in a secure and fully redundant cloud server giving you a 99.9% uptime guarantee.

**Appendix I: ISO3166 Country Code**

This table lists all valid ISO3166 two characters country codes that returns from component API query and describe the country names behind these country codes.

Country Code	Country Name
AD	ANDORRA
AE	UNITED ARAB EMIRATES
AF	AFGHANISTAN
AG	ANTIGUA AND BARBUDA
AI	ANGUILLA
AL	ALBANIA
AM	ARMENIA
AN	NETHERLANDS ANTILLES
AO	ANGOLA
AP	ASIA PACIFIC
AQ	ANTARCTICA
AR	ARGENTINA
AS	AMERICAN SAMOA
AT	AUSTRIA
AU	AUSTRALIA
AW	ARUBA
AZ	AZERBAIJAN
BA	BOSNIA AND HERZEGOWINA
BB	BARBADOS
BD	BANGLADESH
BE	BELGIUM
BF	BURKINA FASO
BG	BULGARIA
BH	BAHRAIN
BI	BURUNDI
BJ	BENIN
BM	BERMUDA
BN	BRUNEI DARUSSALAM
BO	BOLIVIA
BR	BRAZIL
BS	BAHAMAS
BT	BHUTAN
BV	BOUVET ISLAND
BW	BOTSWANA
BY	BELARUS
BZ	BELIZE
CA	CANADA
CC	COCOS (KEELING) ISLANDS
CD	CONGO, THE DEMOCRATIC REPUBLIC OF THE
CF	CENTRAL AFRICAN REPUBLIC
CG	CONGO

Country Code	Country Code
CH	SWITZERLAND
CI	COTE D'IVOIRE
CK	COOK ISLANDS
CL	CHILE
CM	CAMEROON
CN	CHINA
CO	COLOMBIA
CR	COSTA RICA
CS	CZECHOSLOVAKIA (FORMER)
CU	CUBA
CV	CAPE VERDE
CX	CHRISTMAS ISLAND
CY	CYPRUS
CZ	CZECH REPUBLIC
DE	GERMANY
DJ	DJIBOUTI
DK	DENMARK
DM	DOMINICA
DO	DOMINICAN REPUBLIC
DZ	ALGERIA
EC	ECUADOR
EE	ESTONIA
EG	EGYPT
EH	WESTERN SAHARA
ER	ERITREA
ES	SPAIN
ET	ETHIOPIA
EU	EUROPEAN UNION
FI	FINLAND
FJ	FIJI
FK	FALKLAND ISLANDS (MALVINAS)
FM	MICRONESIA, FEDERATED STATES OF
FO	FAROE ISLANDS
FR	FRANCE
FX	FRANCE, METROPOLITAN
GA	GABON
GB	GREAT BRITAIN
GD	GRENADA
GE	GEORGIA
GF	FRENCH GUIANA
GH	GHANA
GI	GIBRALTAR
GL	GREENLAND
GM	GAMBIA
GN	GUINEA
GP	GUADELOUPE
GQ	EQUATORIAL GUINEA
GR	GREECE



Country Code	Country Code
GS	SOUTH GEORGIA & SOUTH SANDWICH ISLANDS
GT	GUATEMALA
GU	GUAM
GW	GUINEA-BISSAU
GY	GUYANA
HK	HONG KONG
HM	HEARD ISLAND AND MCDONALD ISLANDS
HN	HONDURAS
HR	CROATIA
HT	HAITI
HU	HUNGARY
ID	INDONESIA
IE	IRELAND
IL	ISRAEL
IN	INDIA
IO	BRITISH INDIAN OCEAN TERRITORY
IQ	IRAQ
IR	IRAN, ISLAMIC REPUBLIC OF
IS	ICELAND
IT	ITALY
JM	JAMAICA
JO	JORDAN
JP	JAPAN
KE	KENYA
KG	KYRGYZSTAN
KH	CAMBODIA
KI	KIRIBATI
KM	COMOROS
KN	SAINT KITTS AND NEVIS
KP	KOREA, DEMOCRATIC PEOPLE'S REPUBLIC OF
KR	KOREA, REPUBLIC OF
KW	KUWAIT
KY	CAYMAN ISLANDS
KZ	KAZAKSTAN
LA	LAO PEOPLE'S DEMOCRATIC REPUBLIC
LB	LEBANON
LC	SAINT LUCIA
LI	LIECHTENSTEIN
LK	SRI LANKA
LR	LIBERIA
LS	LESOTHO
LT	LITHUANIA
LU	LUXEMBOURG
LV	LATVIA
LY	LIBYAN ARAB JAMAHIRIYA
MA	MOROCCO
MC	MONACO
MD	MOLDOVA, REPUBLIC OF

Country Code	Country Code
MG	MADAGASCAR
MH	MARSHALL ISLANDS
MK	MACEDONIA, THE FORMER YUGOSLAV
ML	MALI
MM	MYANMAR
MN	MONGOLIA
MO	MACAU
MP	NORTHERN MARIANA ISLANDS
MQ	MARTINIQUE
MR	MAURITANIA
MS	MONTSERRAT
MT	MALTA
MU	MAURITIUS
MV	MALDIVES
MW	MALAWI
MX	MEXICO
MY	MALAYSIA
MZ	MOZAMBIQUE
NA	NAMIBIA
NC	NEW CALEDONIA
NE	NIGER
NF	NORFOLK ISLAND
NG	NIGERIA
NI	NICARAGUA
NL	NETHERLANDS
NO	NORWAY
NP	NEPAL
NR	NAURU
NU	NIUE
NZ	NEW ZEALAND
OM	OMAN
PA	PANAMA
PE	PERU
PF	FRENCH POLYNESIA
PG	PAPUA NEW GUINEA
PH	PHILIPPINES
PK	PAKISTAN
PL	POLAND
PM	SAINT PIERRE AND MIQUELON
PN	PITCAIRN
PR	PUERTO RICO
PS	PALESTINIAN TERRITORY, OCCUPIED
PT	PORTUGAL
PW	PALAU
PY	PARAGUAY
QA	QATAR
RE	REUNION
RO	ROMANIA

Country Code	Country Code
RU	RUSSIAN FEDERATION
RW	RWANDA
SA	SAUDI ARABIA
SB	SOLOMON ISLANDS
SC	SEYCHELLES
SD	SUDAN
SE	SWEDEN
SG	SINGAPORE
SH	SAINT HELENA
SI	SLOVENIA
SJ	SVALBARD AND JAN MAYEN
SK	SLOVAKIA
SL	SIERRA LEONE
SM	SAN MARINO
SN	SENEGAL
SO	SOMALIA
SR	SURINAME
ST	SAO TOME AND PRINCIPE
SU	RUSSIAN FEDERATION
SV	EL SALVADOR
SY	SYRIAN ARAB REPUBLIC
SZ	SWAZILAND
TC	TURKS AND CAICOS ISLANDS
TD	CHAD
TF	FRENCH SOUTHERN TERRITORIES
TG	TOGO
TH	THAILAND
TJ	TAJIKISTAN
TK	TOKELAU
TM	TURKMENISTAN
TN	TUNISIA
TO	TONGA
TP	EAST TIMOR
TR	TURKEY
TT	TRINIDAD AND TOBAGO
TV	TUVALU
TW	TAIWAN, PROVINCE OF CHINA
TZ	TANZANIA, UNITED REPUBLIC OF
UA	UKRAINE
UG	UGANDA
UK	UNITED KINGDOM
UM	UNITED STATES MINOR OUTLYING ISLANDS
US	UNITED STATES
UY	URUGUAY
UZ	UZBEKISTAN
VA	HOLY SEE (VATICAN CITY STATE)
VC	SAINT VINCENT AND THE GRENADINES
VE	VENEZUELA

Country Code	Country Code
VG	VIRGIN ISLANDS, BRITISH
VI	VIRGIN ISLANDS, U.S.
VN	VIET NAM
VU	VANUATU
WF	WALLIS AND FUTUNA
WS	SAMOA
YE	YEMEN
YT	MAYOTTE
YU	YUGOSLAVIA
ZA	SOUTH AFRICA
ZM	ZAMBIA
ZW	ZIMBABWE

**Appendix II: ISO4127 Currency Code**

Currency Code	Currency Name
AED	United Arab Emirates dirham
ANG	Netherlands Antillean guilder
ARS	Argentine peso
AUD	Australian dollar
BRL	Brazilian real
BSD	Bahamian dollar
CAD	Canadian dollar
CHF	Swiss franc
CLP	Chilean peso
CNY	Renminbi
COP	Colombian peso
CZK	Czech koruna
DKK	Danish krone
EUR	Euro
FJD	Fijian dollar
GBP	Pound sterling
GHS	Ghana cedi
GTQ	Guatemalan quetzal
HKD	Hong Kong dollar
HNL	Honduran lempira
HRK	Croatian kuna
HUF	Hungarian forint
IDR	Indonesian rupiah
ILS	Israeli new shekel
INR	Indian Rupee
ISK	Icelandic króna
JMD	Jamaican dollar
JPY	Japanese yen
KRW	South Korean won
LKR	Sri Lankan rupee
MAD	Moroccan dirham
MMK	Burmese kyat
MXN	Mexican peso
MYR	Malaysian ringgit
NOK	Norwegian krone
NZD	New Zealand dollar
PAB	Panamanian balboa
PEN	Peruvian nuevo sol
PHP	Philippine peso
PKR	Pakistani Rupee
PLN	Polish złoty
RON	Romanian leu
RSD	Serbian dinar
RUB	Russian ruble
SEK	Swedish krona

SGD	Singapore dollar
THB	Thai baht
TND	Tunisian dinar
TRY	Turkish lira
TTD	Trinidad and Tobago dollar
TWD	New Taiwan dollar
USD	United States dollar
VEF	Venezuelan bolivar
VND	Vietnamese đồng
XAF	Central African CFA franc
XCD	East Caribbean dollar
XPF	CFP franc
ZAR	South African rand

## Appendix III: Sample Code

IP2Currency 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/ip2currencysamplecodes.aspx>

### i. ASP.NET – VB.NET (SOAP)

```
Private Sub IP2CurrencyWebService()  
    Dim x_IP2Currency As New IP2CurrencyWebService  
    Dim oIP2Currency As New IP2CurrencyWebService.IP2CurrencyOutput  
    Dim iIP2Currency As New IP2CurrencyWebService.IP2CurrencyInput  
  
    Try  
        iIP2Currency.VISITORIP = Me.txtVisitorIP.Text  
        iIP2Currency.FROMCURRENCYCODE = Me.txtFromCurrencyCode.Text  
        iIP2Currency.TOCURRENCYCODE = Me.txtToCurrencyCode.Text  
        iIP2Currency.FROMAMOUNT = Me.txtFromAmount.Text  
        iIP2Currency.VISITORIP = Me.txtVisitorIP.Text  
        iIP2Currency.LICENSE = Me.txtLicense.Text  
  
        oIP2Currency = x_IP2Currency.IP2Currency(iIP2Currency)  
  
        txtResult.Text = "FROMCURRENCYCODE:" &  
oIP2Currency.FROMCURRENCYCODE & vbNewLine  
        txtResult.Text += "TOCURRENCYCODE:" & oIP2Currency.TOCURRENCYCODE &  
vbNewLine  
        txtResult.Text += "TOCURRENCYSYMBOL:" &  
oIP2Currency.TOCURRENCYSYMBOL & vbNewLine  
        txtResult.Text += "FROMAMOUNT:" & oIP2Currency.FROMAMOUNT &  
vbNewLine  
        txtResult.Text += "TOAMOUNT:" & oIP2Currency.TOAMOUNT & vbNewLine  
        txtResult.Text += "CONVERSIONRATE:" & oIP2Currency.CONVERSIONRATE &  
vbNewLine  
        txtResult.Text += "TOCOUNTRYNAME:" & oIP2Currency.TOCOUNTRYNAME &  
vbNewLine  
        txtResult.Text += "TOCOUNTRYCODE:" & oIP2Currency.TOCOUNTRYCODE &  
vbNewLine  
        txtResult.Text += "CREDITSAVAILABLE:" &  
oIP2Currency.CREDITSAVAILABLE & vbNewLine  
        txtResult.Text += "MESSAGE:" & oIP2Currency.MESSAGE & vbNewLine  
    Catch ex As Exception  
        Response.Write(ex.Message)  
    End Try  
End Sub
```

**ii. ASP.NET – C#.NET (SOAP)**

```
private void IP2CurrencyWebService()
{
    IP2CurrencyWebService x_IP2Currency = new IP2CurrencyWebService();
    IP2CurrencyOutput oIP2Currency = new IP2CurrencyOutput();
    IP2CurrencyInput iIP2Currency = new IP2CurrencyInput();

    try
    {
        iIP2Currency.VISITORIP = this.txtVisitorIP.Text;
        iIP2Currency.FROMCURRENCYCODE = this.txtFromCurrencyCode.Text;
        iIP2Currency.TOCURRENCYCODE = this.txtToCurrencyCode.Text;
        iIP2Currency.FROMAMOUNT = this.txtFromAmount.Text;
        iIP2Currency.VISITORIP = this.txtVisitorIP.Text;
        iIP2Currency.LICENSE = this.txtLicense.Text;

        oIP2Currency = x_IP2Currency.IP2Currency(iIP2Currency);

        this.txtResult.Text = "FROMCURRENCYCODE:" +
oIP2Currency.FROMCURRENCYCODE + "\n";
        this.txtResult.Text += "TOCURRENCYCODE:" +
oIP2Currency.TOCURRENCYCODE + "\n";
        this.txtResult.Text += "TOCURRENCYSYMBOL:" +
oIP2Currency.TOCURRENCYSYMBOL + "\n";
        this.txtResult.Text += "FROMAMOUNT:" + oIP2Currency.FROMAMOUNT +
"\n";
        this.txtResult.Text += "TOAMOUNT:" + oIP2Currency.TOAMOUNT + "\n";
        this.txtResult.Text += "CONVERSIONRATE:" +
oIP2Currency.CONVERSIONRATE + "\n";
        this.txtResult.Text += "TOCOUNTRYNAME:" +
oIP2Currency.TOCOUNTRYNAME + "\n";
        this.txtResult.Text += "TOCOUNTRYCODE:" +
oIP2Currency.TOCOUNTRYCODE + "\n";
        this.txtResult.Text += "CREDITSAVAILABLE:" +
oIP2Currency.CREDITSAVAILABLE + "\n";
        this.txtResult.Text += "MESSAGE:" + oIP2Currency.MESSAGE + "\n";
    }
    catch (Exception ex)
    {
        Response.Write(ex.Message);
    }
}
```



### iii. PHP

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

    if ($license == "<Enter License Key>" || $license == "")
    {
        echo "license key is a required field." ;
    }
    else
    {
        $wsdl = "http://v1.fraudlabs.com/ip2currencywebservice.asmx?wsdl";

        $client = new SoapClient($wsdl);
        $parms = array("VISITORIP" => $visitorip, "FROMCURRENCYCODE" =>
$fromcurrencycode, "TOCURRENCYCODE" => $tocurrencycode, "FROMAMOUNT" =>
$fromamount, "LICENSE" => $license);

        $result = $client->IP2Currency($parms);

        echo "FROMCURRENCYCODE = " . $result->FROMCURRENCYCODE . "<br>";
        echo "TOCURRENCYCODE = " . $result->TOCURRENCYCODE . "<br>";
        echo "TOCURRENCYSYMBOL = " . $result->TOCURRENCYSYMBOL . "<br>";
        echo "FROMAMOUNT = " . $result->FROMAMOUNT . "<br>";
        echo "TOAMOUNT = " . $result->TOAMOUNT . "<br>";
        echo "CONVERSIONRATE = " . $result->CONVERSIONRATE . "<br>";
        echo "TOCOUNTRYNAME = " . $result->TOCOUNTRYNAME . "<br>";
        echo "TOCOUNTRYCODE = " . $result->TOCOUNTRYCODE . "<br>";
        echo "CREDITSAVAILABLE = " . $result->CREDITSAVAILABLE . "<br>";
        echo "MESSAGE = " . $result->MESSAGE . "<br>";
    }
}
?>
```

### iv. JAVA

```
public synchronized String IP2Currency(String strVISITORIP, String
strFROMCURRENCYCODE, String strTOCURRENCYCODE, String strFROMAMOUNT,
String strLICENSE) throws SOAPException
{
    String returnValue = "";
    if (this.url_ == null)
    {
        throw new SOAPException (Constants.FAULT_CODE_CLIENT, "A URL must
be specified through ApacheSoapProxy.setEndPoint(URL)");
    }
}
```

```
}
// Get this from the soapAction attribute on the
// soap:operation element that is found within the SOAP
// binding information in the WSDL
this.soapActionUri_ = "http://v1.fraudlabs.com/";
ApacheMessageBody ourBody = new ApacheMessageBody ();

// Set the argument
ourBody.strVISITORIP = strVISITORIP;
ourBody.strFROMCURRENCYCODE = strFROMCURRENCYCODE;
ourBody.strTOCURRENCYCODE = strTOCURRENCYCODE;
ourBody.strFROMAMOUNT = strFROMAMOUNT;
ourBody.strLICENSE = strLICENSE;
//Replace the default body with our own
this.envelope_.setBody (ourBody);
message_.send (this.getEndPoint(), this.soapActionUri_,
this.envelope_);
try
{
    //Since the Body.unmarshall() handler is static, we can't
    //replace the basic machinery easily. Instead, we must obtain and
    parse the message on our own.
    this.soapMessage_ = this.message_.receive();
    XMLReader reader =
(XMLReader)Class.forName("org.apache.xerces.parsers.SAXParser").newInst
ance();
    SAXHandler handler = new SAXHandler();
    String strResult = "";

    handler.setElementToSearchFor("FROMCURRENCYCODE");
    reader.setContentHandler(handler);
    reader.parse( new InputSource (new
StringReader( this.soapMessage_.getContent().toString() )));
    strResult += "FROMCURRENCYCODE: " + handler.getResult();

    handler.setElementToSearchFor("TOCURRENCYCODE");
    reader.setContentHandler(handler);
    reader.parse( new InputSource (new
StringReader( this.soapMessage_.getContent().toString() )));
    strResult += "\nTOCURRENCYCODE: " + handler.getResult();

    handler.setElementToSearchFor("TOCURRENCYSYMBOL");
    reader.setContentHandler(handler);
    reader.parse( new InputSource (new
StringReader( this.soapMessage_.getContent().toString() )));
    strResult += "\nTOCURRENCYSYMBOL: " + handler.getResult();

    handler.setElementToSearchFor("FROMAMOUNT");
    reader.setContentHandler(handler);
    reader.parse( new InputSource (new
StringReader( this.soapMessage_.getContent().toString() )));
    strResult += "\nFROMAMOUNT: " + handler.getResult();

    handler.setElementToSearchFor("TOAMOUNT");
    reader.setContentHandler(handler);
    reader.parse( new InputSource (new
StringReader( this.soapMessage .getContent().toString() )));
```

```
        strResult += "\nTOAMOUNT: " + handler.getResult();

        handler.setElementToSearchFor("CONVERSIONRATE");
        reader.setContentHandler(handler);
        reader.parse( new InputSource (new
StringReader( this.soapMessage_.getContent().toString() )));
        strResult += "\nCONVERSIONRATE: " + handler.getResult();

        handler.setElementToSearchFor("TOCOUNTRYNAME");
        reader.setContentHandler(handler);
        reader.parse( new InputSource (new
StringReader( this.soapMessage_.getContent().toString() )));
        strResult += "\nTOCOUNTRYNAME: " + handler.getResult();

        handler.setElementToSearchFor("TOCOUNTRYCODE");
        reader.setContentHandler(handler);
        reader.parse( new InputSource (new
StringReader( this.soapMessage_.getContent().toString() )));
        strResult += "\nTOCOUNTRYCODE: " + handler.getResult();

        handler.setElementToSearchFor("CREDITSAVAILABLE");
        reader.setContentHandler(handler);
        reader.parse( new InputSource (new
StringReader( this.soapMessage_.getContent().toString() )));
        strResult += "\nCREDITSAVAILABLE: " + handler.getResult();

        handler.setElementToSearchFor("MESSAGE");
        reader.setContentHandler(handler);
        reader.parse( new InputSource (new
StringReader( this.soapMessage_.getContent().toString() )));
        strResult += "\nMESSAGE: " + handler.getResult();

        returnValue = strResult;
    }
    catch (Exception exception)
    {
        exception.printStackTrace ();
    }
    return returnValue;
}
```