

IP2Location™ Web Service

~~ Bringing Geography to the Internet ~~

1. Overview	3
1.1 Overview of the IP2Location Web Service	4
1.2 Front End of IP2Location Web Service	5
1.2.1. Integration with In-house System	5
1.3 Back End of IP2Location Web Service	7
1.3.1. IP Address to IP Number Conversion	7
1.3.2. Record Matching	7
1.4 Process Flow Overview	8
2. Implementation	9
2.1 Basic Parameters - Input	10
2.2 Basic Parameters - Output	10
2.3 List of possible value for MESSAGE field	11
3. Design Information	12
3.1 Placement of IP2Location Web Service	12
Appendix I : ISO3166 Country Code	13
Appendix II : Sample Code	19

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 IP2Location Web Services.

For more information, please visit <http://www.fraudlabs.com> or contact your IP2Location representative:

Email : sales@fraudlabs.com

1.1 Overview of the IP2Location Web Service

The IP2Location Web Service consists of 3 separate services that return different data depending on your needs.

The 3 web services in the IP2Location web service are:

- IP2Location Basic Geolocation Premium Web Service
- IP2Location Standard Geolocation Premium Web Service
- IP2Location Professional Geolocation Premium Web Service

IP2Location delivers scalable web services solutions that help you to determine geographical and other information about your online visitors in real time. We are able to identify user's geographical location, such as country, region, city, latitude, longitude, ISP and domain name using proprietary IP address lookup database and technology without invading Internet user's privacy.

IP2Location Web Service is a hosted, programmable XML Web Service that allows exchange of data between systems. Customers can integrate our web service regardless of their web server and other business solutions. This XML web service is used under authorized license to IP2Location.net, the leading geolocation service provider.

IP2Location Web Service is very easy to set up and use. You may understand your visitors better by geographical location within a minute of time. From the information that the web service return to you, you may also control contents distribution by region for digital rights management, or even filter access from countries that you do not do business with to avoid legal issues.

Key Features Include:

- Pinpoints the exact location of customers, with country, region, city, latitude and longitude, Internet Service Providers and domain name using IP address
- Covers all IP address range
- Provides a complete XML-based Web Services API
- Contains sample code examples for ease integration

1.2 Front End of IP2Location 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 IP2Location 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/ip2locationsamplecodes.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/ip2locationwebserviceclientvb.zip>
 - ii. Microsoft ASP.Net (v1.1) - C#:
<http://www.fraudlabs.com/samplecode/ip2locationwebserviceclientcsharp.zip>
 - iii. Microsoft ASP.NET (v2.0) - VB.NET:
<http://www.fraudlabs.com/samplecode/ip2locationwebserviceclientvb2005.zip>
 - iv. Microsoft ASP.Net (2.0) - C#:
<http://www.fraudlabs.com/samplecode/ip2locationwebserviceclientcsharp2005.zip>
 - v. Java:
<http://www.fraudlabs.com/samplecode/ip2locationwebserviceclientjava.zip>
 - vi. PHP:
<http://www.fraudlabs.com/samplecode/ip2locationwebserviceclientphp.zip>
 - vii. Python:
<http://www.fraudlabs.com/samplecode/ip2locationwebserviceclientpython.zip>
 - viii. Perl:
<http://www.fraudlabs.com/samplecode/ip2locationwebserviceclientperl.zip>
 - ix. ColdFusion 9:

<http://www.fraudlabs.com/samplecode/ip2locationwebserviceclientcoldfusion.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 IP2Location™ Geolocation, 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 IP2Location™ Geolocation, 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 IP2Location 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 IP2Location Web Service.

The process works as follows:

1. User submits an IP address that to be analyzed
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 determine the IP address
4. Returns and display the geographical information

1.3.1. IP Address to IP Number Conversion

If the provided IP address is 161.132.13.1, then the IP number is 2709785857. - *How to get this conversion done?*

In general, the formula to convert an IP Address to IP Number is given as below:

Assume that the IP Address is in this format: A.B.C.D.

$$\text{IP Number} = A * (256*256*256) + B * (256*256) + C * (256) + D$$

So for the IP address 161.132.13.1, the IP number is:

$$\begin{aligned} \text{IP Number} &= 161 * (256*256*256) + 132 *(256*256) + 13 * \\ &\quad (256) + 1 \\ &= 2709785857 \end{aligned}$$

1.3.2. Record Matching

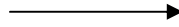
First, we convert the search IP Address to IP Number. Search a record that matches the range condition. It will only give us one match per query. The country information is attached to country fields of the record in our database.

$$\text{IP From} \leq \text{IP Number} \leq \text{IP To}$$

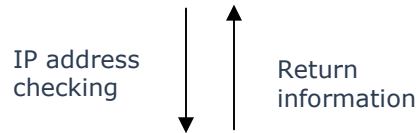
1.4 Process Flow Overview



Step 1
Visitors visit websites



Step 2
Visitor's IP address
captured by merchant



Step 3
IP2Location Web Service
(Geographical information retrieval)

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 IP2Location Web Service.

A WSDL is available at:

<http://v1.fraudlabs.com/ip2locationwebservice.asmx?wsdl>

For more information about IP2Location Web Service implementation, please visit <http://www.fraudlabs.com> or contact your IP2Location representative:

Email: sales@fraudlabs.com

2.1 Basic Parameters - Input

Field	Format	Description
IP	Required	IP address of Internet visitor
LICENSE	Required	License key for free license and premium users.

2.2 Basic Parameters - Output

Field	Versions			Format	Description
	B	S	P		
COUNTRYCODE	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	String	ISO-3166 two character country name of the IP address
COUNTRYNAME	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	String	Country name of the IP address
REGION	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	String	Region name of the IP address.
CITY	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	String	City name of the IP address.
LATITUDE	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Float	Latitude of the IP address.
LONGITUDE	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Float	Longitude of the IP address.
ZIPCODE	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	String	Zip code of the IP address.
ISPNAME	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	String	ISP of the IP address.
DOMAINNAME	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	String	Domain name of the IP address.
TIME_ZONE	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	String	UTC time zone of the IP address.
NETSPEED	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	String	Internet connection type. DIAL = dial up, DSL = broadband/cable, COMP = company/T1
AREACODE	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	String	A varying length number assigned to geographic areas for call between cities.
WEATHERSTATION	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	String	The name of the nearest weather observation station.
WEATHERSTATIONCODE	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	String	The special code to identify the nearest weather observation station.
CREDITSAVAILABLE	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	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	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	String	Web Service Message Response

B - IP2Location Basic Geolocation Premium Web Service

S - IP2Location Standard Geolocation Premium Web Service

P - IP2Location Professional Geolocation Premium Web Service

2.3 List of possible value for MESSAGE field

IP2Location Web Service Error Message
<ul style="list-style-type: none">• Invalid license key
<ul style="list-style-type: none">• Invalid IP address
<ul style="list-style-type: none">• IP address and License Key cannot be blank
<ul style="list-style-type: none">• No credit available
<ul style="list-style-type: none">• License key was expired
<ul style="list-style-type: none">• Not available for this package

3. Design Information

This section provides suggestion regarding the placing of IP2Location 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 IP2Location Web Service

As IP2Location web service provides you very informative details regarding identification of online visitors' location by using IP address, this service can be placed on any forms on your website, offering a seamless integration. In fact, you may already be using this kind of service without realizing it.

Description: Visitors geographical information is needed when visitor attempts to access to certain websites, especially when it involves online transactions.

Level of security: High

Site Type(s):

- e-Commerce Solutions
- Internet Retailers
- Internet Security Specialist
- Fraud Screening and Web Analytics
- Software Developers

Benefits:

- Understand your visitors better by geographical location
- Control contents distribution by region for digital rights management
- Prevent password sharing and abuse of service for information security
- Redirect web pages based on geographical region for load balancing
- Web log statistics and analysis to understand your visitors better by geographical location
- Reduce credit card fraud
- Auto-selection of country on forms
- Filter access from countries you do not do business with to avoid legal issues
- Save advertisement costs by GEO targeting for increased sales and click-through

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	GADELOUPE
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: Sample Code

IP2Location 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/ip2locationsamplecodes.aspx>

i. ASP.NET – VB.NET (SOAP)

```
Private Sub IP2LocationWebService()  
    Dim x_IP2Location As New IP2LocationWebService.IP2LocationWebService  
    Dim oIP2Location As New IP2LocationWebService.IP2LocationOutput  
    Dim iIP2Location As New IP2LocationWebService.IP2LocationInput  
  
    Try  
        iIP2Location.IP = Me.txtIP.Text  
        iIP2Location.LICENSE = Me.txtLicense.Text  
  
        oIP2Location = x_IP2Location.IP2Location(iIP2Location)  
  
        txtResult.Text = "COUNTRYCODE:" & oIP2Location.COUNTRYCODE &  
vbNewLine  
        txtResult.Text += "COUNTRYNAME:" & oIP2Location.COUNTRYNAME &  
vbNewLine  
        txtResult.Text += "REGION:" & oIP2Location.REGION & vbNewLine  
        txtResult.Text += "CITY:" & oIP2Location.CITY & vbNewLine  
        txtResult.Text += "LATITUDE:" & oIP2Location.LATITUDE & vbNewLine  
        txtResult.Text += "LONGITUDE:" & oIP2Location.LONGITUDE & vbNewLine  
        txtResult.Text += "ZIPCODE:" & oIP2Location.ZIPCODE & vbNewLine  
        txtResult.Text += "ISPNAME:" & oIP2Location.ISPNAME & vbNewLine  
        txtResult.Text += "DOMAINNAME:" & oIP2Location.DOMAINNAME &  
vbNewLine  
        txtResult.Text += "TIME_ZONE:" & oIP2Location.TIME_ZONE & vbNewLine  
        txtResult.Text += "NETSPEED:" & oIP2Location.NETSPEED & vbNewLine  
        txtResult.Text += "AREACODE:" & oIP2Location.AREACODE & vbNewLine  
        txtResult.Text += "WEATHERSTATION:" & oIP2Location.WEATHERSTATION &  
vbNewLine  
        txtResult.Text += "WEATHERSTATIONCODE:" &  
oIP2Location.WEATHERSTATIONCODE & vbNewLine  
        txtResult.Text += "CREDITSAVAILABLE:" &  
oIP2Location.CREDITSAVAILABLE & vbNewLine  
        txtResult.Text += "MESSAGE:" & oIP2Location.MESSAGE & vbNewLine  
  
    Catch ex As Exception  
        Response.Write(ex.Message)  
    End Try  
End Sub
```

ii. ASP.NET – C#.NET (SOAP)

```
private void IP2LocationWebService()
{
    IP2LocationWebService x_IP2Location = new IP2LocationWebService();
    IP2LocationOutput oIP2Location = new IP2LocationOutput();
    IP2LocationInput iIP2Location = new IP2LocationInput();

    try
    {
        iIP2Location.IP = this.txtIP.Text;
        iIP2Location.LICENSE = this.txtLicense.Text;

        oIP2Location = x_IP2Location.IP2Location(iIP2Location);

        this.txtResult.Text = "COUNTRYCODE:" + oIP2Location.COUNTRYCODE +
"\n";
        this.txtResult.Text += "COUNTRYNAME:" + oIP2Location.COUNTRYNAME +
"\n";
        this.txtResult.Text += "REGION:" + oIP2Location.REGION + "\n";
        this.txtResult.Text += "CITY:" + oIP2Location.CITY + "\n";
        this.txtResult.Text += "LATITUDE:" + oIP2Location.LATITUDE + "\n";
        this.txtResult.Text += "LONGITUDE:" + oIP2Location.LONGITUDE +
"\n";
        this.txtResult.Text += "ZIPCODE:" + oIP2Location.ZIPCODE + "\n";
        this.txtResult.Text += "ISPNAME:" + oIP2Location.ISPNAME + "\n";
        this.txtResult.Text += "DOMAINNAME:" + oIP2Location.DOMAINNAME +
"\n";
        this.txtResult.Text += "TIME_ZONE:" + oIP2Location.TIME_ZONE +
"\n";
        this.txtResult.Text += "NETSPEED:" + oIP2Location.NETSPEED + "\n";
        this.txtResult.Text += "AREACODE:" + oIP2Location.AREACODE + "\n";
        this.txtResult.Text += "WEATHERSTATION:" +
oIP2Location.WEATHERSTATION + "\n";
        this.txtResult.Text += "WEATHERSTATIONCODE:" +
oIP2Location.WEATHERSTATIONCODE + "\n";
        this.txtResult.Text += "CREDITSAVAILABLE:" +
oIP2Location.CREDITSAVAILABLE + "\n";
        this.txtResult.Text += "MESSAGE:" + oIP2Location.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
{
    $ip = $_POST["ip"];
    $license = $_POST["license"];

    if ($license == "<Enter License Key>" || $license == "")
    {
        echo "license key is a required field." ;
    }
    else
    {
        $wsdl = "http://v1.fraudlabs.com/ip2locationwebservice.asmx?wsdl";
        $client = new SoapClient($wsdl);
        $parms = array("IP" => $ip,"LICENSE" => $license);

        $result = $client->IP2Location($parms);

        echo "COUNTRYCODE = " . $result->COUNTRYCODE . "<br>";
        echo "COUNTRYNAME = " . $result->COUNTRYNAME . "<br>";
        echo "REGION = " . $result->REGION . "<br>";
        echo "CITY = " . $result->CITY . "<br>";
        echo "LATITUDE = " . $result->LATITUDE . "<br>";
        echo "LONGITUDE = " . $result->LONGITUDE . "<br>";
        echo "ZIPCODE = " . $result->ZIPCODE . "<br>";
        echo "ISPNAME = " . $result->ISPNAME . "<br>";
        echo "DOMAINNAME = " . $result->DOMAINNAME . "<br>";
        echo "TIME_ZONE = " . $result->TIME_ZONE . "<br>";
        echo "NETSPEED = " . $result->NETSPEED . "<br>";
        echo "AREACODE = " . $result->AREACODE . "<br>";
        echo "WEATHERSTATION = " . $result->WEATHERSTATION . "<br>";
        echo "WEATHERSTATIONCODE = " . $result->WEATHERSTATIONCODE .
"<br>";
        echo "CREDITSAVAILABLE = " . $result->CREDITSAVAILABLE . "<br>";
        echo "MESSAGE = " . $result->MESSAGE . "<br>";
    }
}
?>
```

iv. JAVA

```
public synchronized String IP2Location(String strIP, 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.strIP = strIP;
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("COUNTRYCODE");
reader.setContentHandler(handler);
reader.parse( new InputSource (new StringReader
(this.soapMessage_.getContent().toString() ));
strResult += "COUNTRYCODE: " + handler.getResult();

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

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

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

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

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

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

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

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

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

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

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

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

handler.setElementToSearchFor("WEATHERSTATIONCODE");
reader.setContentHandler(handler);
reader.parse( new InputSource (new StringReader
(this.soapMessage_.getContent().toString() ));
strResult += "\nWEATHERSTATIONCODE: " + 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;
}
```