

IP2Proxy™ Web Service

~~ Detect Anonymous Proxy by IP Address ~~

| | |
|--|-----------|
| 1. Overview | 3 |
| 1.1 Overview of the IP2Proxy Web Service | 4 |
| 1.2 Front End of IP2Proxy Web Service | 5 |
| 1.2.1. Integration with In-house System | 5 |
| 1.3 Back End of IP2Proxy 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 | 10 |
| 3. Design Information | 11 |
| 3.1 Placement of IP2Proxy Web Service | 11 |
| 4. Appendix I: ISO3166 Country Code | 12 |
| 5. Appendix II: Sample Code | 18 |

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 IP2Proxy Web Services.

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

Email: sales@fraudlabs.com

1.1 Overview of the IP2Proxy Web Service

IP2Proxy Web Service is the proprietary anonymous IP detection service that enables you to check proxy servers for anonymity in order to reduce fraud for online merchants. As sophisticated fraudsters can confuse online merchants by sending them a fake IP address and bypass IP Geolocation clarification along with their requests, IP2Proxy becomes a critical component and offers an additional layer of protection for your business. Merchants can make use of our service in order to automate order process according to the results that we provide.

IP2Proxy Web Service is hosted on redundant servers, redundant Internet connections, and 24x7x365 monitoring. IP2Proxy Web Service is using platform independent XML format to exchange data between systems. All customers can integrate our web service regardless of their web server and shopping cart solutions.

Key Features Include:

- Protect from IP address spoofing
- Verify whether the online buyer is behind anonymous proxy servers
- Prevent and reduce fraud by authenticating online visitors
- Provides a complete XML-based Web Services API
- Contains sample code examples for ease integration

1.2 Front End of IP2Proxy 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 IP2Proxy 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/ip2proxysamplecodes.aspx> 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/ip2proxysamplecode/ip2proxywbserviceclientvb.zip>
 - ii. Microsoft ASP.Net (v1.1) - C#:
<http://www.fraudlabs.com/ip2proxysamplecode/ip2proxywbserviceclientcsharp.zip>
 - iii. Microsoft ASP.NET (v2.0) - VB.NET:
<http://www.fraudlabs.com/ip2proxysamplecode/ip2proxywbserviceclientvb2005.zip>
 - iv. Microsoft ASP.Net (v2.0) - C#:
<http://www.fraudlabs.com/ip2proxysamplecode/ip2proxywbserviceclientcsharp2005.zip>
 - v. Java:
<http://www.fraudlabs.com/ip2proxysamplecode/ip2proxywbserviceclientjava.zip>
 - vi. PHP:
<http://www.fraudlabs.com/ip2proxysamplecode/ip2proxywbserviceclientphp.zip>
 - vii. Python:
<http://www.fraudlabs.com/ip2proxysamplecode/ip2proxywbserviceclientpython.zip>
 - viii. Perl:
<http://www.fraudlabs.com/ip2proxysamplecode/ip2proxywbserviceclientperl.zip>
 - ix. ColdFusion 9:

<http://www.fraudlabs.com/ip2proxysamplecode/ip2proxywbserviceclientcoldfusion.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 IP2Proxy™ Proxy Detection, 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 IP2Proxy™ Proxy Detection, 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 IP2Proxy 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 IP2Proxy Web Service.

The process works as follows:

1. User submits a form containing the parameters (inputs) 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 analyze the parameters
4. The results then will be submitted for IP address converting and anonymity checking process
5. Returns and display the results

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.

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

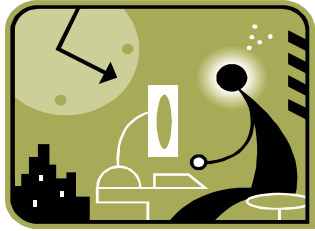
So for the IP address 161.132.13.1, the IP number is:

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

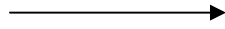
1.3.2. Record Matching

Next, our web service will look through our daily updated database to search a record that matches the anonymous IP address.

1.4 Process Flow Overview



Step 1
Visitors visit websites



Step 2
Visitor's IP address
captured by merchant



Step 3
IP2Proxy Web Service
(Proxy checking)

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 IP2Proxy Web Service.

A WSDL is available at:

<http://v1.fraudlabs.com/ip2proxywebservice.asmx?wsdl>

For more information about IP2Proxy Web Service implementation, please visit <http://www.fraudlabs.com> or contact your IP2Proxy representative:

Email: sales@fraudlabs.com

2.1 Basic Parameters - Input

| Field | Format | Description |
|---------|----------|---|
| IP | Required | IP address of credit card holder. |
| LICENSE | Required | License key for free license and premium users. |

2.2 Basic Parameters - Output

| Field | Format | Description |
|------------------|-----------|--|
| IPADDRESS | String | IP address that to be checked. |
| ANONYMOUSPROXY | YES or NO | Whether IP address is from Anonymous Proxy Server. |
| 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

| IP2Proxy 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 |

3. Design Information

This section provides suggestion regarding the placing of IP2Proxy 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 IP2Proxy Web Service

As IP2Proxy web service provides intensive proxy checking and provides you a very easy and quick description, this service can be placed on ordering form or transaction form on your website, offering a seamless integration.

Description: Proxy checking is 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
- Online Credit Card Merchants
- Web Analytics
- e-Commerce Software Developers

Benefits:

- Protect from IP address spoofing
- Verify whether the online buyer is behind anonymous proxy servers
- Prevent and reduce fraud by authenticating online visitors
- reduces chargeback

For detecting online fraud, we strongly recommend using the FraudLabs Web Service since that it examines an online transaction from various angles with the integration of full and critical modules, and provides fraud score as a guideline.

4. 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 |

5. Appendix II: Sample Code

IP2Proxy 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/ip2proxysamplecodes.aspx>

i. ASP.NET – VB.NET (SOAP)

```
Private Sub IP2ProxyWebService()  
    Dim x_IP2Proxy As New IP2ProxyWebService.IP2ProxyWebService  
    Dim oIP2Proxy As New IP2ProxyWebService.IP2ProxyOutput  
    Dim iIP2Proxy As New IP2ProxyWebService.IP2ProxyInput  
  
    Try  
        iIP2Proxy.IP = Me.txtIP.Text  
        iIP2Proxy.LICENSE = Me.txtLicense.Text  
  
        oIP2Proxy = x_IP2Proxy.IP2Proxy(iIP2Proxy)  
  
        txtResult.Text = "IPADDRESS:" & oIP2Proxy.IPADDRESS & vbNewLine  
        txtResult.Text += "ANONYMOUSPROXY:" & oIP2Proxy.ANONYMOUSPROXY &  
vbNewLine  
        txtResult.Text += "CREDITSAVAILABLE:" & oIP2Proxy.CREDITSAVAILABLE  
& vbNewLine  
        txtResult.Text += "MESSAGE:" & oIP2Proxy.MESSAGE & vbNewLine  
  
        Catch ex As Exception  
            Response.Write(ex.Message)  
        End Try  
    End Sub
```

ii. ASP.NET – C#.NET (SOAP)

```
private void IP2ProxyWebService()  
{  
    IP2ProxyWebService x_IP2Proxy = new IP2ProxyWebService();  
    IP2ProxyOutput oIP2Proxy = new IP2ProxyOutput();  
    IP2ProxyInput iIP2Proxy = new IP2ProxyInput();  
  
    try  
    {  
        iIP2Proxy.IP = this.txtIP.Text;  
        iIP2Proxy.LICENSE = this.txtLicense.Text;  
  
        oIP2Proxy = x_IP2Proxy.IP2Proxy(iIP2Proxy);  
  
        this.txtResult.Text = "IPADDRESS:" + oIP2Proxy.IPADDRESS + "\n";  
        this.txtResult.Text += "ANONYMOUSPROXY:" + oIP2Proxy.ANONYMOUSPROXY  
+ "\n";  
        this.txtResult.Text += "CREDITSAVAILABLE:" +
```

```
oIP2Proxy.CREDITSAVAILABLE + "\n";
    this.txtResult.Text += "MESSAGE:" + oIP2Proxy.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/ip2proxywebservice.asmx?wsdl";
        $client = new SoapClient($wsdl);
        $parms = array("IP" => $ip, "LICENSE" => $license);

        $result = $client->IP2Proxy($parms);

        echo "IPADDRESS = " . $result->IPADDRESS . "<br>";
        echo "ANONYMOUSPROXY = " . $result->ANONYMOUSPROXY . "<br>";
        echo "CREDITSAVAILABLE = " . $result->CREDITSAVAILABLE . "<br>";
        echo "MESSAGE = " . $result->MESSAGE . "<br>";
    }
}
?>
```

iv. JAVA

```
public synchronized String IP2Proxy(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("IPADDRESS");
    reader.setContentHandler(handler);
    reader.parse( new InputSource (new
StringReader( this.soapMessage_.getContent().toString() )));
    strResult += "IPADDRESS: " + handler.getResult();

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