INTRODUCTION

IP2Location™ web service is providing a service to reverse lookup of IP address to an ISO3166 country code, region name, city name, latitude, longitude, ZIP code, ISP, and domain. This document explains in details the semantics of the URL function calls you can make to use this web service for demonstration or commercial purposes. In this document, you will learn:

- How to install, setup and test the IP2Location™ Web API
- How IP2Location™ web service query syntax works
- How to interpret the search results from IP2Location™ Web API

If you have comments or just have questions, please contact sales@ip2location.com.

SEARCH REQUEST & RESPONSE FORMAT

Search requests submit a query string and a set of parameters to the IP2Location™ web service and receive a search result. Search results are derived from IP2Location™ index of over 4 billion unique IP addresses.

2.1 Search Parameters

This table lists all the valid name-value pairs that can be used in a search request and describes how these parameters will modify the search results.

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ip</td>
<td>(required) IP address for reverse lookup purpose. Both IPv4 and IPv6 are supported. If not present, the server IP address will be used for the reverse lookup.</td>
</tr>
<tr>
<td>key</td>
<td>(required) API key.</td>
</tr>
<tr>
<td>package</td>
<td>(optional) If not present, the web service will assume the WS1 package query. Valid value: WS1</td>
</tr>
<tr>
<td></td>
<td>Values returned for WS1: Country Code (2-digits)</td>
</tr>
<tr>
<td></td>
<td>Values returned for WS5: Country Name, Country Code (2-digits), Region, City, Latitude, Longitude</td>
</tr>
<tr>
<td></td>
<td>Values returned for WS9: Country Name, Country Code (2-digits), Region, City, Latitude, Longitude, ZIP Code</td>
</tr>
<tr>
<td></td>
<td>Values returned for WS10: Country Name, Country Code (2-digits), Region, City, Latitude, Longitude, ZIP Code, ISP, Domain</td>
</tr>
<tr>
<td>format</td>
<td>(optional) If not present, text format will be returned as the search result. Valid value: json</td>
</tr>
</tbody>
</table>
Notes: Different credits will be consumed for different package call. Please refer to http://www.ip2location.com/web-service for details.

2.2 Search Web API
IP2Location™ web service uses a single web interface to perform the reverse lookup logic. The protocol used is HTTP or HTTPS GET. You can test this API as easy as using a web browser.

Basic syntax:
http://api.ip2location.com/?ip=<ipadd>&key=<key>

Search with different package (default is WS1)
http://api.ip2location.com/?ip=<ipadd>&key=<key>&package=<package>

Search with different search result format (default is text)
http://api.ip2location.com/?ip=<ipadd>&key=<key>&package=<package>&format=<format>
(Note: if you need a text format return, do not enter the &format into the query string)

API Examples:
WS1 Query: http://api.ip2location.com/?ip=<ip_address>&key=<your_api_key>
WS1 Response: US

WS5 Query: http://api.ip2location.com/?ip=<ip_address>&key=<your_api_key>&package=WS5
WS5 Response: US;UNITED STATES;CALIFORNIA;MOUNTAIN VIEW;37.405992;-122.078515

WS9 Query: http://api.ip2location.com/?ip=<ip_address>&key=<your_api_key>&package=WS9
WS9 Response: US;UNITED STATES;CALIFORNIA;MOUNTAIN VIEW;37.405992;-122.078515;94043

WS10 Query: http://api.ip2location.com/?ip=<ip_address>&key=<your_api_key>&package=WS10
WS10 Response: US;UNITED STATES;CALIFORNIA;MOUNTAIN VIEW;37.405992;-122.078515;94043;GOOGLE INCORPORATED;GOOGLE.COM
SEARCH RESULT STATUS

All successful HTTP request will return a response code of 200. In our web API, it also returns the lookup country code if successful or error code if failed.

3.1 Error Codes

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ACCOUNT_MAX_LIMIT</td>
<td>Account has not enough credit to perform the lookup. Users need to top-up the account before can use the service again.</td>
</tr>
<tr>
<td>ACCOUNT_EXPIRED</td>
<td>Account has been expired. Users need to renew the license before can use the service again.</td>
</tr>
<tr>
<td>DEMO_MAX_LIMIT</td>
<td>Maximum attempts reached for demo version. The demo version allows 20 queries per day from an IP address. You could try the demo again from the same IP address after our server refresh once per day.</td>
</tr>
<tr>
<td>INVALID_ACCOUNT</td>
<td>Invalid combination of account ID or password used.</td>
</tr>
<tr>
<td>INVALID_IP_ADDRESS</td>
<td>Invalid IP address provided in query string.</td>
</tr>
<tr>
<td>MISSING_ACC</td>
<td>Account ID is missing from query string.</td>
</tr>
<tr>
<td>MISSING_IP</td>
<td>IP address is missing from query string.</td>
</tr>
<tr>
<td>MISSING_PASS</td>
<td>Password is missing from query string.</td>
</tr>
</tbody>
</table>

AUTO RECHARGE

User can configure the auto recharge on the license page to allow the system to automatically top up the credits when it falls below the 10% threshold. Users will be required to enter the payment information before enabling the recharge option.

Note: The system performs the threshold check in every 10 minutes interval. Therefore, the recharge maybe not happening immediately upon hitting the threshold, but within 10 minutes period.
SETUP AND INSTALLATION

This section demonstrates how to set up and to use the API from server-side scripts such as ASP and PHP. However, it does not limit to the example listed here. Any other programming languages or scripting languages that support HTTP should be able to use this web API.

4.1 Active Server Pages

```vbscript
Dim objHttp, strQuery
strQuery = "http://api.ip2location.com/?" & _
"ip=" & ipAddress & _
"&key=" & key
set objHttp = Server.CreateObject("Msxml2.ServerXMLHTTP")
objHttp.open "GET", strQuery, false
objHttp.send
Response.Write objHttp.ResponseText
Set objHttp = Nothing
```

Requirements:
1. ASP 3.0+
2. Microsoft® XML 3.0 Component

Microsoft® XML 3.0 Component is free to be downloaded at http://msdn.microsoft.com/xml/general/xmlparser.asp

4.2 VB.NET

```vbnet
Imports System.Net
Imports System.IO

Private Sub WebAPI()
    Dim strQuery As String
    Dim IPAddress As String = "218.111.156.5"
    Dim Key As String = "demo"
    Dim HttpWReq As HttpWebRequest
    Dim HttpWResp As Net.HttpWebResponse

    strQuery = "http://api.ip2location.com/?" & _
    "ip=" & IPAddress & _
    "&key=" & Key

    HttpWReq = WebRequest.Create(strQuery)
    HttpWReq.Method = "GET"
    HttpWResp = HttpWReq.GetResponse()
    Dim reader As System.IO.StreamReader = New IO.StreamReader(HttpWResp.GetResponseStream())
    Response.Write(reader.ReadToEnd)
End Sub
```

Requirements:
1. Microsoft® .NET Framework 1.1 and above
4.3 C#

```csharp
using System.Net;
using System.IO;

private void WebAPI()
{
    string strQuery;
    string IPAddress = "218.111.156.5";
    string key = "demo";

    HttpWebRequest HttpWReq;
    HttpWebResponse HttpWResp;
    strQuery = "http://api.ip2location.com/?" +
               "ip=" + IPAddress +
               
```

```
               
```
               
```
               "&key=" + key;

    HttpWReq = (HttpWebRequest)WebRequest.Create(strQuery);
    HttpWReq.Method = "GET";
    HttpWResp = (HttpWebResponse)HttpWReq.GetResponse();
    System.IO.StreamReader reader = new System.IO.StreamReader(HttpWResp.GetResponseStream());
    Response.Write(reader.ReadToEnd());
}
```

Requirements:
1. Microsoft® .NET Framework 1.1 and above

4.4 PHP: Hypertext Preprocessor

```php
$url = parse_url($query);
$host = $url["host"];  
$path = $url["path"] . 
$fp = fsockopen ($host, 80, $errno, $errstr, 60) or die('Can not open 
connection to server.');
if (!$fp) {
    echo "$errstr ($errno)<br>
```
4.5 Perl/CGI

```perl
use LWP::UserAgent;
use HTTP::Request::Common qw(GET);

my $ipaddress = $ENV{"REMOTE_ADDR"};
my $query = "http://api.ip2location.com/?ip=" . $ipaddress . "&acc=" . $key;
my $ua = new LWP::UserAgent;
my $res = $ua->request(GET $query);
if ($res->is_success())
{
    my $co = $res->content;
}
print $co;
```

4.4 Web Browser

Copy URL below into a web browser. Replace all parameters with the correct values.

http://api.ip2location.com/?ip=<ipadd>&key=<key>