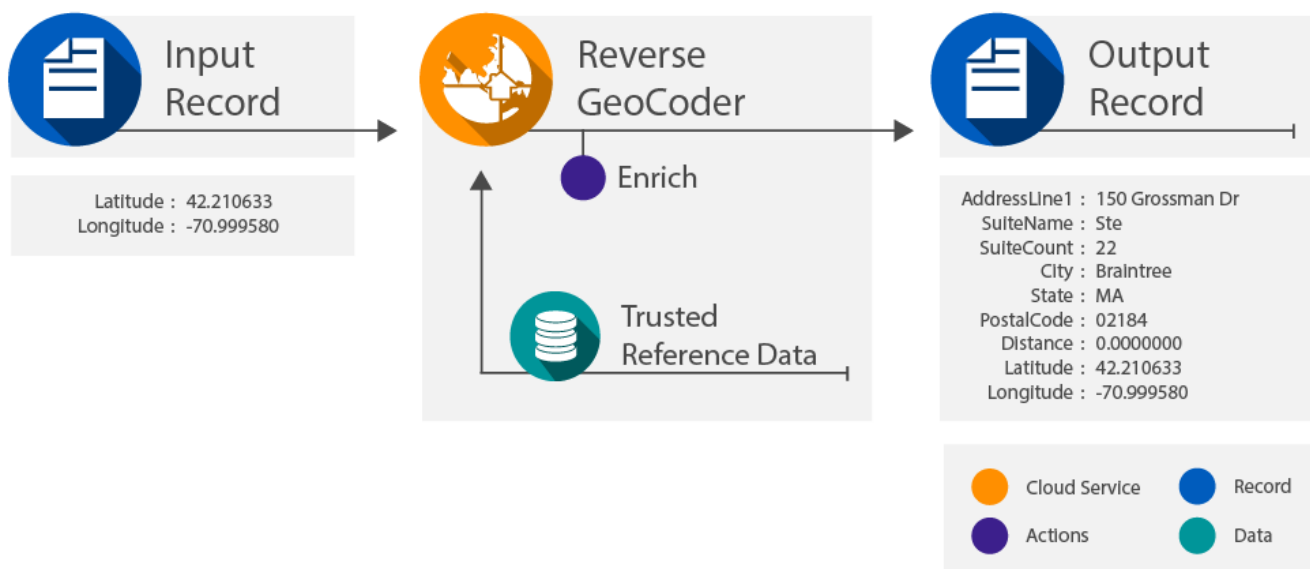


Reverse GeoCoder Service: Programmer's Quick Start

Overview

Reverse GeoCoder from Melissa Data is a Web Service that provides the nearest valid U.S addresses to a latitude and longitude coordinate. Reverse GeoCoder converts a geographic coordinate to a valid street address that can be used, for example, in mobile apps to locate a person in case of an emergency or to provide roadside help. It also can be used by mailers to create a targeted mailing list assembled from addresses grouped around a specific geographic location.

Reverse GeoCoder is ideal for vehicle tracking, for location-based services, and other applications where you have a GPS signal with latitude and longitude and want information about the address or place this location represents.



You can use Reverse GeoCoder to:

- Create a mailing list within a specific radius for targeted marketing.
- Use in mobile apps for emergency location purposes & provide roadside help.

Reverse GeoCoder has the ability to:

- Return suite name & number of suites in each address.
 - Return up to 100 addresses for each Lat/Long coordinate.
 - Support multiple protocols including JSON, XML, and REST.
-

FIELDS INPUT AND OUTPUT FROM THE SERVICE

INPUTs	Description
Transmission Reference	A unique string value identifying the request
Customer ID	License String from Melissa Data
Latitude	The latitude geographic coordinate in decimal format.
Longitude	The longitude geographic coordinate in decimal format.
MaxDistance	Optional. The requested maximum distance (in miles). If no MaxDistance is specified, the default is 10 miles.

OUTPUTs	Description
Version	This is a string value that is the current revision number of Reverse GeoCoder.
Transmission Reference	A unique string value identifying the request
TransmissionResults	This is a string value that lists error codes related to the webservice, request structure, or your customerID.
Results	This is a string value with comma delimited status and error codes for the submitted record.
TotalRecords	This is a string value that is the total number of records returned with the response.
AddressLine1	Returns the street address (street number and street name) that corresponds to the input geo coordinates.
SuiteName	Returns the suite name of the address returned. Values that could be returned are: Ste, Apt., etc.
SuiteCount	Returns the number of suites in a particular building. Zero will be returned if the address has a no suites (a single delivery point).
City	Returns the city of the output address.
State	Returns the State of the output address.
PostalCode	Returns the Postal Code of the output address.
AddressKey	Returns a unique identifier for an address. This key can be used with other current and future Melissa Data services. The AddressKey is an eleven digit number: ZIP5+Plus4+Delivery Point (2 digits)
Latitude	Returns the latitude geographic coordinate of the output address.
Longitude	Returns the longitude geographic coordinate of the output address.
Distance	Returns the distance between the input coordinates and the output coordinates. A distance of zero indicates that an exact address match was returned for the input geographic coordinates.

License String

You should have been provided an encrypted and unique license string or Customer ID from Melissa. This must be included with each request to the Reverse GeoCoder Service. This value should be put into the CustomerID element in each Web service request.

If you do not have a license string, please contact your Melissa sales representative at 1-800-MELISSA (1-800-635-4772).

Sample REST Requests

1. <http://reversegeo.melissadata.net/V3/WEB/ReverseGeoCode/doLookup?id=12345678&lat=42.210633&long=-70.999580%20&dist=0.001&format=json>

Note: The default output format is XML (the format variable can be omitted to output XML.)

Sample JSON Response

```
{
  "Version":"3.0.0.6",
  "TransmissionReference":"sample",
  "TransmissionResults": "",
  "Results":"GS07",
  "TotalRecords":1,
  "Records":[{"
    "AddressLine1":"150 Grossman Dr",
    "SuiteName":"Ste","SuiteCount":"22",
    "City":"Braintree",
    "State":"MA",
    "PostalCode":"02184",
    "AddressKey":"02184495299",
    "Latitude":"","Longitude":"",
    "Distance":"0.000000"}
  ]
}
```

Sample XML Response

```
<ResponseArray>
  <Version>3.0.0.6</Version>
  <TransmissionReference>sample</TransmissionReference>
  <TransmissionResults/>
  <Results>GS07</Results>
  <TotalRecords>1</TotalRecords>
  <Records>
    <ResponseRecord>
      <AddressLine1>150 Grossman Dr</AddressLine1>
      <SuiteName>Ste</SuiteName>
      <SuiteCount>22</SuiteCount>
      <City>Braintree</City>
      <State>MA</State>
      <PostalCode>02184</PostalCode>
      <AddressKey>02184495299</AddressKey>
```

```
<Latitude/><Longitude/>
<Distance>0.000000</Distance>
</ResponseRecord>
</Records>
</ResponseArray>
```

Single vs. Batch

Reverse GeoCoder takes only one record at a time. In other words each request has only one set of geo coordinates (latitude and longitude), but it does return an array of responses if multiple valid addresses are at that particular location.

Reverse GeoCoder Service URLs [Reverse GeoCoder Service Endpoint URLs](#)

Choosing a Web Service Protocol

The Melissa Data Reverse GeoCoder Service supports REST, JSON, and XML. For the undecided, here are some Pros and Cons of one Web Service protocol over the other.

REST

Pros: REST is lightweight and relies upon HTTP to do its work. If you don't need a strict API definition, this is the way to go. REST is also format-agnostic so you can use XML or JSON as responses.

Cons: REST can only be used for sending of single records and doesn't support strict contracts or more involved security. The Response is an XML or JSON document.

XML

Pros: XML allows record set structures of more than one record at a time and has very good support with most languages and browsers. Supports namespaces.

Cons: Developers need to use tools to serialize/de-serialize the XML structure.

JSON

Pros: JSON relies on simple object serialization based on JavaScript's object initialization. It is very simple to use with JavaScript and easily parsed and understood by developers.

Cons: No support for formal definitions. No namespace support. Not much support in Web Service clients with some platforms.

Basic Order of Operations (Pseudo Code)

1. Choose XML or the REST service.
 2. Create an instance of the request object.
 3. Populate the request element CustomerID with your Product License.
 4. Set the options <MaxRecords> and <MaxDistance>
 5. Add input <Latitude> and <Longitude> coordinates in <RequestRecord> element. (XML)
 6. Call the method and pass in the request to the service using the WEB endpoint for XML or JSON requests.
 7. Examine and parse the response from the reply object back from the service.
 8. Interpret the results.
-

Interpreting Results

Melissa Data's Reverse GeoCoder Service uses Results Codes to detail whether returning an address was possible or not.

Melissa Cloud Services use the following Results conventions:

1. CLOUD SERVICE ERRORS: SExx
2. CLOUD TRANSMISSION ERRORS: GExx
3. REVERSE GEOCODER STATUS CODES: GSxx
4. REVERSE GEOCODER ERROR CODES: GExx

For Example: A GS07 Result Code means that returning an address for the given input was successful.

Please check the documentation for any additional information on Results.

Results Codes

The service returns a series of results codes to tell you whether returning an address was possible or not, and why.

For a full list of the Results Codes returned by the Reverse GeoCoder Service, see [Reverse GeoCoder Result Codes](#).

Sample Code

Fully working examples are available on the wiki pages:

[Click here to go to the Reverse GeoCoder Service Sample Code Wiki Page.](#)

Wiki Page

A product support Wiki is available for your convenience. In the Wiki, you will find documentation about the service in more detail.

[Click here to go to the Reverse GeoCoder Service Wiki Page.](#)

Misc. Considerations

Firewall

If you are behind a firewall, you may need to allow specific IP addresses access in order to communicate with the service. For a full list of IP Addresses, see [IP Address Information](#).