**Geo Service Api Documenation**

**Descritipon :**

Returns all none GEOgraphical lookup data for the anewluv API , such as cities, countries postal codes etc

**Help URL :**

<http://173.160.122.195/Anewluv.Web.GeoService/GeoService.svc/Rest/help>

**AuthRequired: Api Key**

a)Sample of api key and authentication header using fiddler : **460ad6f3-8216-469f-9b1c-52cffa5d812c**

you can use the temporary api key for development of :

User-Agent: Fiddler

Content-Type: text/json

Host: 173.160.122.195

apikey: **460ad6f3-8216-469f-9b1c-52cffa5d812c**

b) Sample of Ajax calls with header included:

$.ajax({

                type: "GET",

                url: geoserviceurl + "getfilteredcitiesbycountryandfilter/" + country + "/" + filter,

                contentType: 'application/json',

                xhrFields: {

                    withCredentials: false

                },

                beforeSend: function (xhr) {

                    xhr.setRequestHeader('apikey', '460ad6f3-8216-469f-9b1c-52cffa5d812c');

                },

                headers: {

                },

                success: function (data) {

                    //  self.Employees(data); //Put the response in ObservableArray

                    // return new listitem(data);

                    // var obj = jQuery.parseJSON(data);

                    //debugger

                    if (data) {

                        self.cities(data);

                        self.isOpen(true);

                    }

                },

                error: function (error) {

                    alert(error.status + "<--and--> " + error.statusText);

                }

            });

**Service Call Details and Documenation**

The following details of each call shows the properties required for each call on a call by call basis.

**Service Call details**

1)**getcountryandpostalcodestatuslist** - gets the curren list of all countries allowed and the postal code status of the countries as well as state provinince

**JsonResult**

[{

"code":"String content",

"customregionid":2147483647,

"haspostalcode":true,

"id":"String content",

"name":"String content",

"region":"String content"

}]

Notable members

Code – country code

Haspostalcode – is yes if the country is a country with postal codes available and false otherwise

Id- country id in the database

Name – display name of the country to be seen in dropdowns and used in searches

Region- region in the world , ie AF is Africa nd EU is erourpe , NA – north America , SA – south america

**Help url** :

<http://173.160.122.195/Anewluv.Web.GeoService/GeoService.svc/Rest/help/operations/getcountryandpostalcodestatuslist>

**Url** :

<http://173.160.122.195/Anewluv.Web.GeoService/GeoService.svc/Rest/getcountryandpostalcodestatuslist/>

Result:

[{"code":"AF ","customregionid":null,"haspostalcode":false,"id":null,"name":"Afghanistan","region":"ME "},{"code":"AS ","customregionid":null,"haspostalcode":false,"id":null,"name":"American Samoa","region":"OC "},{"code":"AD ","customregionid":null,"haspostalcode":true,"id":null,"name":"Andorra","region":"EU "},

ETC ….

2)**getcitystateprovincelist** – gets a comma delimited list of cites and state provinces

**JsonResult**

[{

"citystateprovincevalue":"String content",

"selected":true

}]

Notable values

Citystateprovincevalue – the first part of this result is the city (value to store) and the second part is the state or province (used to help diffrentiale values that have similar names in the result)

Example : **Minneapolis**,Kansas

The city name is Minneapolis

**Help url** :

<http://173.160.122.195/Anewluv.Web.GeoService/GeoService.svc/Rest/help/operations/getfilteredcitiesbycountryfilter>

**Url** :

<http://173.160.122.195/Anewluv.Web.GeoService/GeoService.svc/Rest/getfilteredcitiesbycountryandfilter>

Sample call :

<http://173.160.122.195/Anewluv.Web.GeoService/GeoService.svc/Rest/getfilteredcitiesbycountryandfilter>

JSON body

[{

"countryname":"UnitedStates",

"filter":"Minn"

}]

Result:

[{"citystateprovincevalue":"Minneapolis,Kansas","selected":false},{"citystateprovincevalue":"Minneapolis,Minnesota","selected":false},{"citystateprovincevalue":"Minneapolis,North Carolina","selected":false},{"citystateprovincevalue":"Minneola,Florida","selected":false},{"citystateprovincevalue":"Minneola,Kansas","selected":false},{"citystateprovincevalue":"Minneota,Minnesota","selected":false},{"citystateprovincevalue":"Minnesota City,Minnesota","selected":false},{"citystateprovincevalue":"Minnesota Lake,Minnesota","selected":false},{"citystateprovincevalue":"Minnetonka,Minnesota","selected":false},{"citystateprovincevalue":"Minnetonka Beach,Minnesota","selected":false},{"citystateprovincevalue":"Minnewaukan,North Dakota","selected":false}]

3)**getpostalcodelist**– gets a list of postal codes for a country with HASPOSTALCODES= true , using the countryname ,city and a filter string

**JsonResult**

[{

"postalcodevalue":"String content",

"selected":true

}]

Notable values

postalcodevalue – the postal code value

**Help url** :

<http://173.160.122.195/Anewluv.Web.GeoService/GeoService.svc/Rest/help/operations/getpostalcodesbycountrycityfilter>

**Url** :

<http://173.160.122.195/Anewluv.Web.GeoService/GeoService.svc/Rest/getpostalcodesbycountrycityfilter>

Sample call :

<http://173.160.122.195/Anewluv.Web.GeoService/GeoService.svc/Rest/getpostalcodesbycountrycityfilter>

JSON body

[{

"countryname":"UnitedStates",

"city" : "Minneapolis",

"filter":"554"

}]

Result:

[{"postalcodevalue":"55414","selected":false},{"postalcodevalue":"55413","selected":false},{"postalcodevalue":"55412","selected":false},{"postalcodevalue":"55411","selected":false},{"postalcodevalue":"55410","selected":false},{"postalcodevalue":"55409","selected":false},{"postalcodevalue":"55408","selected":false},{"postalcodevalue":"55407","selected":false},{"postalcodevalue":"55406","selected":false},{"postalcodevalue":"55405","selected":false},{"postalcodevalue":"55404","selected":false},{"postalcodevalue":"55403","selected":false},{"postalcodevalue":"55402","selected":false},{"postalcodevalue":"55401","selected":false},{"postalcodevalue":"55473","selected":false},{"postalcodevalue":"55449","selected":false},{"postalcodevalue":"55448","selected":false},{"postalcodevalue":"55434","selected":false},{"postalcodevalue":"55433","selected":false},{"postalcodevalue":"55432","selected":false}]

4)**getgpsdatabycountryandcitystateprovince**– gets a list of gps data (latttitude and longitude) using county and city

**JsonResult**

[{

"lattitude":1.26743233E+15,

"longitude":1.26743233E+15,

"postalcode":"String content",

"selected":true,

"stateprovince":"String content"

}]

Notable values

lattitude – the latitude to save

longitude – the longitude to save

**Help url** :

<http://173.160.122.195/Anewluv.Web.GeoService/GeoService.svc/Rest/help/operations/getgpsdatalistbycountrycitystateprovince>

**Url** :

<http://173.160.122.195/Anewluv.Web.GeoService/GeoService.svc/Rest/getgpsdatalistbycountrycitystateprovince>

Sample call :

<http://173.160.122.195/Anewluv.Web.GeoService/GeoService.svc/Rest/getgpsdatalistbycountrycitystateprovince>

notes – pass the state province as well if it is available from the city list

**stateprovince is optional**

JSON body

[{

"countryname":"UnitedStates",

**“stateprovince”:”Minnesota”,**

"city" : "Minneapolis"

}]

Result:

[{"lattitude":39.129501342773437,"longitude":-97.6688003540039,"postalcode":null,"selected":false,"stateprovince":"Kansas"},{"lattitude":45.052299499511719,"longitude":-93.254096984863281,"postalcode":null,"selected":false,"stateprovince":"Minnesota"},{"lattitude":45.095001220703125,"longitude":-93.2396011352539,"postalcode":null,"selected":false,"stateprovince":"Minnesota"},{"lattitude":45.164299011230469,"longitude":-93.319297790527344,"postalcode":null,"selected":false,"stateprovince":"Minnesota"}]

5)**getgpsdatabycountryandcity**– gets a list of gps data (latttitude and longitude) using county and city

**JsonResult**

[{

"lattitude":1.26743233E+15,

"longitude":1.26743233E+15,

"postalcode":"String content",

"selected":true,

"stateprovince":"String content"

}]

Notable values

lattitude – the latitude to save

longitude – the longitude to save

**Help url** :

<http://173.160.122.195/Anewluv.Web.GeoService/GeoService.svc/Rest/help/operations/getgpsdatalistbycountrycitypostalcode>

**Url** :

<http://173.160.122.195/Anewluv.Web.GeoService/GeoService.svc/Rest/getgpsdatalistbycountrycitypostalcode>

Sample call :

<http://173.160.122.195/Anewluv.Web.GeoService/GeoService.svc/Rest/getgpsdatalistbycountrycitypostalcode>

notes – pass the state province as well if it is available from the city list

**city is optional**

JSON body

[{

"countryname":"UnitedStates",

“postalcode”:”55411”,

**"city" : "Minneapolis"**

}]

Result:

[{"lattitude":45.052299499511719,"longitude":-93.254096984863281,"postalcode":null,"selected":false,"stateprovince":"Minnesota**"}**]

5)**getgpsdatabycountryandcity**

[http://173.160.122.195/GeoService/GeoService.svc/Rest/verifyorupdateregistrationgeodata](http://173.160.122.195/Shell.MVC2.Web.GeoService/GeoService.svc/Rest/verifyorupdateregistrationgeodata)

5)**validategeodata – use this call to validate the country,city and postal code as well as latitude and longtudue at the same time**

<http://173.160.122.195/Anewluv.Web.GeoService/GeoService.svc/Rest/verifyorupdateregistrationgeodata/>