



Developing a Browser geocoder using REST API and ASP.NET

Bhavdeep S Sachdev

The Browser Geocoder



[Home](#) [About](#) [Contact](#)

CDPH Batch Geocoder 2.0 Beta

Welcome to the Batch Geocoder 2.0.

Prerequisites

Upload File

Select WorkSheet

Select Fields

Preview Data

Options

Geocoding

This is still in Beta, if you have any problems, please contact the GIS team. Thank you.

Prerequisites:

- Use only Excel 2010 or higher with extension .xlsx
- First row in the worksheet must be column names
- *Address* field is required in the source worksheet. *Primary Key*, *City*, *State* and *Zip Code* fields are optional.
- A maximum of 50,000 rows will be geocoded
- Maximum file size is 10MB
- There should be no worksheet in the file with the name "BatchGeocoderResults".

Next

The Browser Geocoder



[Home](#) [About](#) [Contact](#)

CDPH Batch Geocoder 2.0 Beta

Welcome to the Batch Geocoder 2.0.

Prerequisites **Upload File** Select WorkSheet Select Fields Preview Data Options Geocoding

Upload an excel file

You start by uploading an excel file with your addresses.
NOTE: Do not upload more than 50,000 addresses.

Upload an excel file to geocode..

AUG14DEL.xlsx

The Browser Geocoder



[Home](#) [About](#) [Contact](#)

CDPH Batch Geocoder 2.0 Beta

Welcome to the Batch Geocoder 2.0.

Prerequisites > Upload File > Select WorkSheet > **Select Fields** > Preview Data > Options > Geocoding

Match the Fields

Please select the fields from your worksheet and match them to the ones required by the batch geocoder.
Only the "Address" field is required, the rest are optional.

Select Columns from the drop down lists

Primary Key

Address *Required. This can be a full address or just the street part of an address.

City

State

Zip Code

[Previous](#) [Next](#)

The Browser Geocoder

Prerequisites

Upload File

Select WorkSheet

Select Fields

Preview Data

Options

Geocoding

Preview your data

Please preview the top 10 rows of data that will be used by the geocoder. If things don't look right, please go back to make changes. If you are satisfied with the data preview, click Next.

21 rows of data found. Displaying the top 10 rows:

ID1	Address	City	Zip
1756	12150 Woodside Ave.	Lakeside	92040
1855	472-103 Johnstonville Rd., North	Susanville	96130-8746
1866	704-795 Bangham Ln.	Susanville	96130-7716
1867	436-965 Susan Dr.	Doyle	96109-0007
1874	472-013 Johnstonville Rd., North	Susanville	96130-8752
3019	9300 Imperial Hwy.	Downey	90242-2813
3067	12721 South Willowbrook Ave.	Compton	90222-1529
3638	3243 East Avenue R-8	Palmdale	93550
4085	46655 Road 200	O'Neals	93645
4096	3200 Pat Ave.	Mojave	93501-1344

Previous

Next

The Browser Geocoder

Prerequisites > Upload File > Select WorkSheet > Select Fields > Preview Data > **Options** > Geocoding

Options

Select the Geocoding Service to use:

- Composite with PointAddress and StreetAddress only
- Composite with PointAddress, StreetAddress, StreetName, Postal, AdminPlaces

Select the additional fields that you want in the output:

<input checked="" type="checkbox"/> Census 2010 Tracts	<input type="checkbox"/> CA Senate District
<input type="checkbox"/> Census 2010 Block Groups	<input type="checkbox"/> CA Senate District Representative Name
<input checked="" type="checkbox"/> Training Resource Centers	<input type="checkbox"/> CA Assembly District
<input checked="" type="checkbox"/> County	<input type="checkbox"/> CA Assembly District Representative Name
<input type="checkbox"/> MSA ID	<input type="checkbox"/> US Congressional Districts
<input type="checkbox"/> MSA Name	<input type="checkbox"/> US Congressional District Representative Name

[Previous](#) [Next](#)

The Browser Geocoder

Prerequisites

Upload File

Select WorkSheet

Select Fields

Preview Data

Options

Geocoding

Geocoding

Please click on the "Start Geocoding" button to start the process.

After the geocoding is complete, the page will refresh with a log of what happened and a "Download File" button will appear. You can click on the "Download File" button to download the results file.

The output excel file will have an additional worksheet named "BatchGeocoderResults" with the following fields:

Your original fields - [*Primary Key, Address, City, State, Zip*]

And the result fields - BatchGeocoderID, X, Y, Locator Name, Status, Score, Matching Address

Geocoding... Please wait...

12:42:46.013 PM - Creating temporary worksheet

12:42:46.399 PM - Starting to Geocode

12:42:55.831 PM - Geocoding complete

12:42:55.833 PM - Adding Overlay fields

Creating Overlay fields... Please wait... 4.76 %. Total Time: 11 second(s)



The Browser Geocoder

Prerequisites

Upload File

Select WorkSheet

Select Fields

Preview Data

Options

Geocoding

Geocoding

Please click on the "Start Geocoding" button to start the process.

After the geocoding is complete, the page will refresh with a log of what happened and a "Download File" button will appear. You can click on the "Download File" button to download the results file.

The output excel file will have an additional worksheet named "BatchGeocoderResults" with the following fields:

Your original fields - [Primary Key, Address, City, State, Zip]

And the result fields - BatchGeocoderID, X, Y, Locator Name, Status, Score, Matching Address

Download Results

Geocode another file

12:42:46.013 PM - Creating temporary worksheet

12:42:46.399 PM - Starting to Geocode

12:42:55.831 PM - Geocoding complete

12:42:55.833 PM - Adding Overlay fields

12:43:05.968 PM - Overlay complete

12:43:05.970 PM - Finalizing...

12:43:05.972 PM - DONE!!!

Geocoding complete in 19 second(s)

Total Records - 21

Successfully Geocoded - 7

Accuracy:

Point Address - 3

Street Address - 4

Street Name - 0

Postal - 0

Admin Places - 0

The REST Endpoint



The screenshot shows a web browser window with the following content:

- Browser tab: GeocodingResources/USA x
- Address bar: gis-apps.cdpn.ca.gov Search Google or type URL
- Page title: ArcGIS REST Services Directory
- Breadcrumbs: Home > services > GeocodingResources > USA CompositeAll (GeocodeServer)
- Links: [JSON](#) | [SOAP](#)
- Section header: **GeocodingResources/USA_CompositeAll (GeocodeServer)**
- Section header: **Service Description:**
- Section header: **Address Fields:**
 - Address (type: esriFieldTypeString , alias: Address , required: false , length: 100)
 - City (type: esriFieldTypeString , alias: City , required: false , length: 100)
 - State (type: esriFieldTypeString , alias: State , required: false , length: 50)
 - Zip (type: esriFieldTypeString , alias: Zip , required: false , length: 50)
- Section header: **Single Line Address Field:**
 - SingleLine (type: esriFieldTypeString , alias: Single Line Input , required: false , length: 100)
- Section header: **Candidate Fields:**
 - Loc_name (type: esriFieldTypeString , alias: Loc_name , required: false , length: 14)
 - Shape (type: esriFieldTypeGeometry , alias: Shape , required: false)
 - Score (type: esriFieldTypeDouble , alias: Score , required: false)
 - Match_addr (type: esriFieldTypeString , alias: Match_addr , required: false , length: 120)
 - Side (type: esriFieldTypeString , alias: Side , required: false , length: 1)
 - AddNum (type: esriFieldTypeString , alias: AddNum , required: false , length: 12)
 - StPreDir (type: esriFieldTypeString , alias: StPreDir , required: false , length: 20)

The REST Endpoint



The screenshot shows a web browser window with the following elements:

- Browser Tab:** Geocode Addresses: (Geo x)
- Address Bar:** gis-apps.cdph.ca.gov Search Google or type URL
- Page Title:** ArcGIS REST Services Directory
- Breadcrumbs:** Home > services > GeocodingResources > USA_CompositeAll (GeocodeServer) > geocodeAddresses
- Section Header:** Geocode Addresses: (GeocodingResources/USA_CompositeAll)
- Form Fields:**
 - Addresses:** A large text input area for entering addresses.
 - Output Spatial Reference:** An empty text input field.
 - Format:** A dropdown menu currently set to HTML.
- Buttons:** Two buttons at the bottom: "Geocode Addresses (GET)" and "Geocode Addresses (POST)".

The REST Endpoint - json

```
//Sample JSON for Geocoding
StringBuilder sb = new StringBuilder();
sb.AppendLine(@"{ ""records"": [");
sb.AppendLine(@"    {");
sb.AppendLine(@"        ""attributes"": {");
sb.AppendLine(@"            ""OBJECTID"": 1,"");
sb.AppendLine(@"            ""Address"": ""1616 Capitol Ave"",");
sb.AppendLine(@"            ""City"": ""Sacramento"",");
sb.AppendLine(@"            ""State"": ""CA"",");
sb.AppendLine(@"            ""Zip"": ""95814""");
sb.AppendLine(@"        }");
sb.AppendLine(@"    },");
sb.AppendLine(@"    {");
sb.AppendLine(@"        ""attributes"": {");
sb.AppendLine(@"            ""OBJECTID"": 2,"");
sb.AppendLine(@"            ""Address"": ""4000 J Street"",");
sb.AppendLine(@"            ""City"": ""Sacramento"",");
sb.AppendLine(@"            ""State"": ""CA""");
sb.AppendLine(@"        }");
sb.AppendLine(@"    }");
sb.AppendLine(@"]");
sb.AppendLine(@"}");
txtGeocoderJSONInput.Text = sb.ToString();
```

The REST Endpoint – request/response

```
//Build Post data
StringBuilder postData = new StringBuilder();
postData.Append("addresses=" + HttpUtility.UrlEncode(txtGeocoderJSONInput.Text) + "&");
postData.Append("outSR=4326&");
postData.Append("f=pjson");

string postDataString = postData.ToString();

//Send the Post request
HttpWebRequest webRequest = (HttpWebRequest)WebRequest.Create(url);
webRequest.Method = "POST";
webRequest.ContentType = "application/x-www-form-urlencoded";
webRequest.ContentLength = postDataString.Length;

using (StreamWriter requestWriter = new StreamWriter(webRequest.GetRequestStream()))
{
    requestWriter.Write(postDataString);
}

//Read the response
string responseData = string.Empty;

using (StreamReader responseReader = new StreamReader(webRequest.GetResponse().GetResponseStream()))
{
    responseData = responseReader.ReadToEnd();
}
```

The REST Endpoint - Output

Input JSON for Geocoding:

```
{ "records": [
  {
    "attributes": {
      "OBJECTID": 1,
      "Address": "1616 Capitol Ave",
      "City": "Sacramento",
      "State": "CA",
      "Zip": "95814"
    }
  },
  {
    "attributes": {
      "OBJECTID": 2,
      "Address": "4000 J Street",
      "City": "Sacramento",
      "State": "CA"
    }
  }
]
```

Submit to Geocoding Service

Output JSON from Geocoder:

```
{
  "spatialReference": {
    "wkid": 4326,
    "latestWkid": 4326
  },
  "locations": [
    {
      "address": "1616 Capitol Ave, Sacramento, Cal",
      "location": {
        "x": -121.48594163399849,
        "y": 38.574701897575608
      },
      "score": 100,
      "attributes": {
        "ResultID": 1,
        "Loc_name": "PointAddress",
        "Status": "M",
        "Score": 100,
        "Match_addr": "1616 Capitol Ave, Sacramento,",
        "Side": "R",

```



DEMO

Reading Excel Spreadsheets

```
oledbConn = new OleDbConnection(@"Provider=Microsoft.ACE.OLEDB.12.0; Data Source=" +
    hdnUploadedFileNameWithPath.Value + "; Extended Properties='Excel 12.0;HDR=YES;IMEX=1;';");
oledbConn.Open();

DataTable dt = oledbConn.GetOleDbSchemaTable(System.Data.OleDb.OleDbSchemaGuid.Tables, null);
bool bOutputSheetFound = false;
for (int i = 0; i < dt.Rows.Count; i++)
{
    string sTemp = dt.Rows[i]["TABLE_NAME"].ToString();
    if (sTemp == "BatchGeocoderResults")
        bOutputSheetFound = true;
}

if (bOutputSheetFound)
{
    UploadStatusLabel.Text = "You cannot have any worksheets with the name \"BatchGeocoderResult\"";
    UploadStatusLabel.ForeColor = System.Drawing.Color.Red;
    oledbConn.Close();
    return;
}

ddlWorksheets.Items.Clear();

ddlWorksheets.DataSource = dt.DefaultView;
ddlWorksheets.DataTextField = "TABLE_NAME";
ddlWorksheets.DataValueField = "TABLE_NAME";
ddlWorksheets.DataBind();
ListItem li = new ListItem("--Select Worksheet--", "-1");
ddlWorksheets.Items.Insert(0, li);
```

Read columns and preview data

```
oledbConn.Open();
```

```
OleDbCommand cmd = new OleDbCommand(); ;  
OleDbDataAdapter oleda = new OleDbDataAdapter();  
DataSet ds = new DataSet();
```

```
cmd.Connection = oledbConn;  
cmd.CommandType = CommandType.Text;  
cmd.CommandText = "SELECT top 1 * FROM [" + ddlWorksheets.SelectedValue + "];"  
oleda = new OleDbDataAdapter(cmd);  
oleda.Fill(ds, "dsS1no");
```

```
sCommandText = "";  
sCommandText = "SELECT top 10 ";  
if (bIDColumn) sCommandText += "[" + ddlObjectID.SelectedValue + "], ";  
sCommandText += "[" + ddlAddress.SelectedValue + "];"  
if (bCityColumn) sCommandText += ", [" + ddlCity.SelectedValue + "];"  
if (bStateColumn) sCommandText += ", [" + ddlState.SelectedValue + "];"  
if (bZipColumn) sCommandText += ", [" + ddlZip.SelectedValue + "];"  
sCommandText += " FROM [" + ddlWorksheets.SelectedValue + "];"
```


Define fields in Output Sheet

```
string sCommand = "Create table [BatchGeocoderResults] ";
sCommand += " (BatchGeocoderID number,";
if (ddlObjectID.SelectedValue != "-1")
    sCommand += " [" + ddlObjectID.SelectedValue + "] string,";

sCommand += " [" + ddlAddress.SelectedValue + "] string,";

if (ddlCity.SelectedValue != "-1")
    sCommand += " [" + ddlCity.SelectedValue + "] string,";
if (ddlState.SelectedValue != "-1")
    sCommand += " [" + ddlState.SelectedValue + "] string,";
if (ddlZip.SelectedValue != "-1")
    sCommand += " [" + ddlZip.SelectedValue + "] string,";
sCommand += " X number, Y number, [Locator Name] string, Status string, Score number,
for (int j = 0; j < selectedLI.Count; j++)
{
    sCommand += ",[" + selectedLI[j].Text + "] string";
}
}
```

Start Geocoding

```
AddToGeocodingLog("\n" + DateTime.Now.ToString("hh:mm:ss.fff tt") + " - Starting to Geocode");
```

```
sqlCmd.CommandType = CommandType.StoredProcedure;  
sqlCmd.CommandText = "UpdateGeocodingLog";  
sqlCmd.Parameters.Add("BatchID", SqlDbType.Int).Value = Convert.ToInt32(hdnBatchGeocoderID.Value);  
sqlCmd.Parameters.Add("StringToAdd", SqlDbType.VarChar, 1000).Value = sStringToAdd;
```

```
if (iTotalNumberOfRecords < 1000)  
    iNumberOfRecordsToGeocodeAtOnce = 50;  
else if (iTotalNumberOfRecords < 5000)  
    iNumberOfRecordsToGeocodeAtOnce = 100;  
else if (iTotalNumberOfRecords < 20000)  
    iNumberOfRecordsToGeocodeAtOnce = 200;
```

Request/Response

```
// create the POST request
HttpWebRequest webRequest = (HttpWebRequest)WebRequest.Create(url);
webRequest.Method = "POST";
webRequest.ContentType = "application/x-www-form-urlencoded";
webRequest.ContentLength = postDataString.Length;

using (StreamWriter requestWriter2 = new StreamWriter(webRequest.GetRequestStream()))
{
    requestWriter2.Write(postDataString);
}

// This actually does the request and gets the response back
HttpWebResponse resp = (HttpWebResponse)webRequest.GetResponse();

string responseData = string.Empty;

using (StreamReader responseReader = new StreamReader(webRequest.GetResponse().GetResponseStream()))
{
    // dumps the HTML from the response into a string variable
    responseData = responseReader.ReadToEnd();
}

dynamic outputObject = JsonConvert.DeserializeObject(responseData);

UpdateResultsSheet(outputObject);
```

Parsing the Response Object

```
for (int i = 0; i < outputObject.locations.Count; i++)
{
    cmd = new OleDbCommand();
    cmd.Connection = oledbConn;
    cmd.CommandType = CommandType.Text;
    cmd.CommandText = @"UPDATE [BatchGeocoderResults] SET"
        + " X = " + outputObject.locations[i].location.x + ", "
        + " Y = " + outputObject.locations[i].location.y + ", "
        + " [Locator Name] = '" + outputObject.locations[i].attributes.Loc_name + "', "
        + " Status = '" + outputObject.locations[i].attributes.Status + "', "
        + " Score = " + outputObject.locations[i].attributes.Score + ", "
        + " [Matching Address] = '" + outputObject.locations[i].attributes.Match_addr + "' "
        + " WHERE BatchGeocoderID = " + outputObject.locations[i].attributes.ResultID + ";";
    cmd.ExecuteNonQuery();

    iSuccessfullyGeocoded++;
}
```

Add Overlay Output

```
using (StreamReader responseReader = new StreamReader(webRequestOverlay.GetResponse().GetResponseStream())
{
    // dumps the HTML from the response into a string variable
    responseDataOverlay = responseReader.ReadToEnd();
}
outputOverlayObject = JsonConvert.DeserializeObject(responseDataOverlay);

UpdateResultsSheetWithOverlayData(outputOverlayObject);

for (int i = 0; i < outputObject.results[0].value.features.Count; i++)
{
    string sSelectedOptions = "";
    for (int j = 0; j < selectedLI.Count; j++)
    {
        if (sSelectedOptions != "")
            sSelectedOptions += ",";
        string sTemp = outputObject.results[0].value.features[i].attributes[selectedLI[j].Value];
        sSelectedOptions += " [" + selectedLI[j].Text + "] = '" + sTemp.Replace("'", "'') + "' ";
    }


    cmd = new OleDbCommand();
    cmd.Connection = oledbConn;
    cmd.CommandType = CommandType.Text;
    cmd.CommandText = @"UPDATE [BatchGeocoderResults] SET"
        + sSelectedOptions
        + " WHERE BatchGeocoderID = " + outputObject.results[0].value.features[i].attributes.
    cmd.ExecuteNonQuery();
}
```

Final Output

D	E	F	G	H	I	J	K	L
City	Zip	X	Y	Locator Name	Status	Score	Matching Address	Census 2010 Tracts
Lakeside	92040	-116.9302927	32.85619511	USA_PointAddress	M	100	0 Woodside Ave, Lakeside, California, 9	06073016702
Sanville	96130-8746	0	0			0		
Sanville	96130-7716	0	0			0		
Doyle	96109-0007	0	0			0		
Sanville	96130-8752	0	0			0		
Downey	90242-2813	-118.1286025	33.9169532	USA_PointAddress	M	100	0 Imperial Hwy, Downey, California, 90	06037551800
Compton	90222-1529	-118.2322575	33.91725951	USA_StreetAddress	M	100	Willowbrook Ave, Compton, California	06037541400
Palmdale	93550	-118.0708835	34.56560323	USA_StreetAddress	M	100	3 E Avenue R 8, Palmdale, California, 93	06037910605
Palmdale	93645	0	0			0		
Mojave	93501-1344	-118.1897983	35.05005902	USA_StreetAddress	M	100	3200 Pat Ave, Mojave, California, 93501	06029005900
Temecula	95965	0	0			0		
Temecula	93645	0	0			0		
Temecula	92592	-117.0952345	33.49804217	USA_StreetAddress	M	100	Meadows Pkwy, Temecula, California,	06065043264
Temecula	95207	0	0			0		
Temecula	95215-9113	0	0			0		
Palo Alto	94303	0	0			0		
Huntington Beach	92647	-118.0127373	33.73405412	USA_PointAddress	M	100	Glen Dr, Huntington Beach, California, 9	06059099605
Temecula	94954-2301	0	0			0		
Temecula	95401-4035	0	0			0		
Temecula	95361	0	0			0		
Temecula	95321-9338	0	0			0		

Future Enhancements

- Run as a Windows service
- Publish as an asynchronous service
- Publish and consume other services on the REST endpoint



```
protected void End_Of_Presentation()  
{  
    Console.WriteLine("Any Questions");  
}
```


Contact

- Bhavdeep S Sachdev
- Bhavdeep.Sachdev@gmail.com

- Get the code:
- <https://github.com/bhavdeep-sachdev/CDPHBrowserGeocoder>