
jquery-geocodify Documentation

Release 0.3

Los Angeles Times Data Desk

June 12, 2014

1	Features	3
2	Example	5
3	Documentation	7
3.1	How it works	7
3.2	Demonstrations	8
3.3	Changelog	11
3.4	Credits	11
4	Contributing	13

Autocomplete for address searches

Features

- Morph an empty form into a search box ready to auto-complete address searches using Google's geocoder
- Handle the user's selected location however you'd like by specifying a simple function
- Bias the results using Google's built-in viewport and region biasing
- Tweak search strings before they're sent and filter geocoder results before they appear

Example

Documentation

3.1 How it works

Import all the dependencies in your page's head

```
<script src="http://ajax.googleapis.com/ajax/libs/jquery/1.11.0/jquery.min.js"></script>
<script type="text/javascript" src="http://maps.googleapis.com/maps/api/js?sensor=false"></script>

<!-- These next two you should host by yourself in production -->
<!-- Download the latest version here https://github.com/datadesk/jquery-geocodify/releases -->
<link rel="stylesheet" src="/path/to/jquery.geocodify.css" />
<script type="text/javascript" src="/path/to/jquery.geocodify.min.js"></script>
```

Add an empty form to your page's body and initialize it with instructions for what function to run when the user selects an address

```
<input id="geocoder" placeholder="Enter a location ..."></input>
<script type="text/javascript">
  $("#geocoder").geocodify({
    onSelect: function (result) { alert(result); }
  });
</script>
```

The example above will just alert the selected address in the browser. It's a JavaScript representation of what is returned by the [Google Maps geocoder](#).

3.1.1 Options

onSelect

A function that takes the Google geocoder's result object and decides what to do with it, like it load it on a map, or redirect to another page, or whatever you need. Required.

acceptableAddressTypes

A whitelist of address types allowed to appear in the results. Drawn from the set defined by Google's geocoder. Optional. All types accepted by default.

errorHandler

A function for handling errors returned by the Google geocoder. Optional. Null by default.

filterResults

A function for filtering results before they appear in the dropdown. Optional. Null by default.

minimumCharacters

Sets the number of characters that must be entered before the geocoder starts to automatically run. Optional. The default is five.

noResultsText

The text that appears when a search returns no results. Optional. The default is “No results found. Please refine your search.”

prepSearchString

A function that treats the search string before it is passed to the geocoder. Optional. Null by default.

regionBias

Instruct the geocoder to return results biased towards a particular region of the world. More information about the available codes can be found [here](#). Optional. Null by default.

viewportBias

Instruct the geocoder to return results biased towards a bounding box presented in Google’s data format. Google’s documentation can be found [here](#). Optional. Null by default.

3.2 Demonstrations

Examples of jquery-geocodify in action. *Full documentation is elsewhere.*

3.2.1 The basic box

```
<input id="geocodify-basic-box"></input>
<script type="text/javascript">
  $("#geocodify-basic-box").geocodify({
    onSelect: function (result) { alert(result); }
  });
</script>
```

3.2.2 Address type whitelisting

The whitelist of acceptable address types can be used to filter the results before they appear in the dropdown. In this example, the input is configured to only return airports. Try searching for “LAX” or “Charles De Gaulle.”

```
<input id="geocodify-address-type-whitelisting"></input>
<script type="text/javascript">
  $("#geocodify-address-type-whitelisting").geocodify({
    onSelect: function (result) { alert(result); },
    acceptableAddressTypes: [
      'airport'
    ],
    minimumCharacters: 3
  });
</script>
```

3.2.3 Filter results

Geocoder results can be filtered before they appear in the dropdown by passing in a function. It should accept a list of Google geocoder objects and return whatever list you’d like to keep. This example drops any results that aren’t filed in Los Angeles County.

```

<input id="geocodify-filter-results"></input>
<script type="text/javascript">
  $("#geocodify-filter-results").geocodify({
    onSelect: function (result) { alert(result); },
    filterResults: function(results) {
      // Using some underscore.js tricks here to filter faster
      // http://underscorejs.org/
      return _.filter(results, function (addy) {
        return _.some(addy.address_components, function (component) {
          return component.long_name === 'Los Angeles County';
        });
      });
    }
  });
</script>

```

3.2.4 Initial text

You can provide a string to load when the box first appears.

```

<input id="geocodify-initial-text" placeholder="Enter an address"></input>
<script type="text/javascript">
  $("#geocodify-initial-text").geocodify({
    onSelect: function (result) { alert(result); },
  });
</script>

```

3.2.5 No results text

You can provide a string for the dropdown when no results return. Try searching some nonsense like “qwerty.”

```

<input id="geocodify-no-results-text"></input>
<script type="text/javascript">
  $("#geocodify-no-results-text").geocodify({
    onSelect: function (result) { alert(result); },
    noResultsText: "Nein!"
  });
</script>

```

3.2.6 Minimum characters

Sets the number of characters that must be entered before the geocoder starts to automatically run. This example reduces the number to 2. Try searching “LAX.”

```

<input id="geocodify-minimum-characters"></input>
<script type="text/javascript">
  $("#geocodify-minimum-characters").geocodify({
    onSelect: function (result) { alert(result); },
    minimumCharacters: 2
  });
</script>

```

3.2.7 Prep search string

A function that treats the search string before it is passed to the geocoder. This example adds “California” to the search if the user has not provided it.

```
<input id="geocodify-prep-search-string"></input>
<script type="text/javascript">
  $("#geocodify-prep-search-string").geocodify({
    onSelect: function (result) { alert(result); },
    prepSearchString: function (query) {
      var pattr = /\sca\s|\scalifornia\s/gi;
      var match = query.match(pattr);
      if (!match) {
        return query + ' California';
      } else {
        return query;
      }
    }
  });
</script>
```

3.2.8 Region bias

Instruct the geocoder to return results biased towards a particular region of the world. More information about the available codes can be found [here](#). This example biases results to Spain.

```
<input id="geocodify-region-bias"></input>
<script type="text/javascript">
  $("#geocodify-region-bias").geocodify({
    onSelect: function (result) { alert(result); },
    regionBias: "ES"
  });
</script>
```

3.2.9 Viewport bias

Instruct the geocoder to return results biased towards a bounding box presented in Google’s data format. Google’s documentation can be found [here](#). This example biases results to a box surrounding Los Angeles County.

```
<input id="geocodify-viewport-bias"></input>
<script type="text/javascript">
  $("#geocodify-viewport-bias").geocodify({
    onSelect: function (result) { alert(result); },
    viewportBias: new google.maps.LatLngBounds(
      new google.maps.LatLng(33.22030778968541,-118.948974609375),
      new google.maps.LatLng(35.0120020431607,-117.44384765625)
    )
  });
</script>
```

3.3 Changelog

3.3.1 0.2.0

- CSS refactored and not included in a separate file from the JavaScript

3.3.2 0.1.0

- Initial release

3.4 Credits

This library was created by [Ben Welsh](#) of the [Los Angeles Times Data Desk](#). Valuable contributions have been made by [albertsun](#) and [unruthless](#). Inspiration was provided by the [Chicago Tribune News Applications team](#).

Contributing

- Code repository: <https://github.com/datadesk/jquery-geocodify>
- Issues: <https://github.com/datadesk/jquery-geocodify/issues>