|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Date:** | **06/6/2020** | **Name:** | **Namratha S Hipparagi** | |
| **Course:** | **Python** | **USN:** | **4al16ec040** | |
| **Topic:** | **Application 11: Project Exercise on Building a Geocoder Web service** | **Semester & Section:** | **8 A** | |
| **AFTERNOON SESSION DETAILS** | | | |
| REPORT  **Geocoding API** is enabled in the Google Cloud Platform Console, in the same project you set up for the Maps JavaScript API. **Geocoding** is the process of converting addresses (like "1600 Amphitheatre Parkway, Mountain View, CA") into geographic coordinates (like latitude 37.423021 and longitude -122.083739), which you can use to place markers or position the map. Before using the Geocoding service in the Maps JavaScript API, first ensure that the  The GeocoderRequest object literal contains the following fields:  {  address: string,  location: LatLng,  placeId: string,  bounds: LatLngBounds,  componentRestrictions: GeocoderComponentRestrictions,  region: string }   1. Go to the [Google Cloud Platform Console](https://console.cloud.google.com/project/_/apiui/apis/enabled). 2. Click the **Select a project** button, then select the same project you set up for the Maps JavaScript API and click **Open**. 3. From the list of APIs on the **Dashboard**, look for **Geocoding API**. 4. If you see the API in the list, you’re all set. If the API is not listed, enable it:    1. At the top of the page, select **ENABLE API** to display the **Library** tab. Alternatively, from the left side menu, select **Library**.    2. Search for **Geocoding API**, then select it from the results list.    3. Select **ENABLE**. When the process finishes, **Geocoding API** appears in the list of APIs on the **Dashboard**.   The GeocoderResult object represents a single geocoding result. A geocode request may return multiple result objects:  results[]: {  types[]: string,  formatted\_address: string,  address\_components[]: {    short\_name: string,    long\_name: string,    postcode\_localities[]: string,    types[]: string  },  partial\_match: boolean,  place\_id: string,  postcode\_localities[]: string,  geometry: {    location: LatLng,    location\_type: GeocoderLocationType    viewport: LatLngBounds,    bounds: LatLngBounds  } }   * 1. **Required parameters:** You must supply one, and only one, of the following fields: address — The address which you want to geocode. location — The LatLng (or LatLngLiteral) for which you wish to obtain the closest, human-readable address. The geocoder performs a reverse geocode. See [Reverse Geocoding](https://developers.google.com/maps/documentation/javascript/geocoding#ReverseGeocoding) for more information. placeId — The place ID of the place for which you wish to obtain the closest, human-readable address.   **Optional parameters:** bounds — The [LatLngBounds](https://developers.google.com/maps/documentation/javascript/reference/coordinates" \l "LatLngBounds) within which to bias geocode results more prominently. The bounds parameter will only influence, not fully restrict, results from the geocoder.   * componentRestrictions — Used to restrict results to a specific area. See more information about [component filtering](https://developers.google.com/maps/documentation/javascript/geocoding#ComponentFiltering) below. * region — The region code, specified as a [IANA language region subtag](http://www.iana.org/assignments/language-subtag-registry). In most cases, these tags map directly to familiar ccTLD ("top-level domain") two-character values. The region parameter will only influence, not fully restrict, results from the geocoder. See more information about [region code biasing](https://developers.google.com/maps/documentation/javascript/geocoding#GeocodingRegionCodes) below. | | | |