Google Maps API

Team Details

- 180010001 - Abhinav Singh
- 180010010 - Sai Yashwanth
- 180030042 - Utkarsh Prakash
- 180010038 - Tarun Goyal

Google Map API

Google Maps API helps us to integrate Google Maps with web pages and mobile applications.

Google Map is a web mapping service developed by Google in 2005. It is estimated that Google Maps is used by 1 billion users per month in 2020. Google Maps API can be used to access satellite images, aerial photography, real time traffic conditions, getting driving directions (fastest), finding nearby restaurants, finding nearby ATMs etc.

API Key

An API (Application Programming Interface) key is usually a long string that we can include in the request URL.

API Key is used as a unique identifier for authentication, to know who is accessing the API. API key might authorize a user to allow certain actions. Users can put restrictions on api keys such as hostname (API key can’t be accessed from outside the given domain) etc.

Creating Google Map API Key

1. Create a project in the link:

   https://console.cloud.google.com/projectselector2/home/dashboard
2. Go to **APIs & Services**.
3. In that, Go to the **Credentials** page.
4. Then, Create Credentials -> **API Key**.
5. The new API key is listed on the **Credentials** page under **API keys**.
6. You can also **restrict** the API also.

Adding API Key to your request URL :

You must include an API key with every Directions API request. In the following example, replace YOUR_API_KEY with your API key.

—

https://maps.googleapis.com/maps/api/directions/json?origin=Toronto&destination=Montreal&key=YOUR_API_KEY

Terminologies

- Latitude
- Longitude
- Geocoding
- Reverse Geocoding
- Geolocation
- Types of Maps

Latitude

Latitude is a geographic coordinate that specifies the north–south position of a point on the Earth's surface.

Longitude

Longitude, is a geographic coordinate that specifies the east–west position of a point on the Earth's surface.

- The Earth is divided into degrees of longitude and latitude which helps us measure location and time using a single standard.
- Basically latitudes and longitudes help in understanding the view of the Earth.
- We can find a place by using latitude and longitude.
- We can compare the times at different places using latitude and longitude.

Geocoding

Geocoding is the process of converting addresses (like a street address) into geographic coordinates (like latitude and longitude). This is mainly used in Google Maps. When you search a place in the search bar and when we select the place, the geocoding will be done and we will find the place by getting its latitude and longitude.

Reverse Geocoding

Reverse geocoding is the process of converting geographic coordinates (like latitude and longitude) into a human-readable address. This is also mainly used in Google Maps. When you enter the latitude and longitude of an address, it will give the place where that is located.

Geolocation

Geolocation collects its data from the GPS (Global Positioning System) on your device and cellular networks. Since devices are used by individuals, geolocation uses positioning systems to track an individual's whereabouts down to latitude and longitude coordinates, or more practically, a physical address. Both mobile and desktop devices can use geolocation.

Types of Maps

The following map types are available in the Maps JavaScript API:

- `roadmap` displays the default road map view. This is the default map type.
- `satellite` displays Google Earth satellite images.
- `hybrid` displays a mixture of normal and satellite views.
- `terrain` displays a physical map based on terrain information.

JSFIDDLE

It is an online IDE which we are using to execute our code. JSFiddle provide us a free openly available Google map api key which can only be accessed on their hostname.

HTML

```html
<!DOCTYPE html>
<html>
  <head>
    <title>Simple Map</title>
    <script src="https://polyfill.io/v3/polyfill.min.js?features=default"></script>
    <script
      src="https://maps.googleapis.com/maps/api/js?key=AIzaSyBIwzALxUPNbatRBj3Xi1Uhp0fFzwWNBkE&callback=initMap&libraries=&v=weekly"
      defer
    ></script>
    <!-- jsFiddle will insert css and js -->
  </head>
  <body>
    <div id="map"></div>
  </body>
</html>
```

JavaScript + No-Library (pure JS)

```javascript
let map;

function initMap() {
  map = new google.maps.Map(document.getElementById("map"), {
    center: { lat: -34.397, lng: 150.644 },
    zoom: 8,
  });
}
```

CSS

Tidy

```css
/* Always set the map height explicitly to define the size of the div
       * element that contains the map. */
#map {
  height: 100%;
}

/* Optional: Makes the sample page fill the window. */
html,
body {
  height: 100%;
  margin: 0;
  padding: 0;
}
```

Console (beta)

URL for calling Google Maps API

A typical URL for loading the Maps JavaScript API:

```
https://maps.googleapis.com/maps/api/js?key=YOUR_API_KEY
&callback=FUNCTION_NAME
&v=VERSION
&libraries="LIBRARIES"
&language="LANGUAGE"
&region="REGION"
```

The script tag loads the Maps JavaScript API:

```
<script defer
    src="https://maps.googleapis.com/maps/api/js?key=YOUR_API_KEY&cal
    lback=initMap">
</script>
```

Places Library

Used for following purposes : -

- Find Place from Query
- Find Place from Phone Number
- Text Search
- Nearby Search
- Place Details Requests

Find Place from Query

Method used :
findPlaceFromQuery()

Parameters :-

- Query
- Fields
- LocationBias

Find Place from Phone Number

Method used :
findPlaceFromPhoneNumber()

Parameters :-

- phoneNumber
- Fields
- LocationBias (Optional)

Nearby Search Requests

Method used :
nearbySearch()

Parameters :-

- Location
- minPriceLevel (Optional)
- maxPriceLevel (Optional)
- openNow (Optional)
- rankBy (Optional)

Text Search Requests

Method used :
textSearch()

Parameters :-

- Query
- openNow (Optional)
- minPriceLevel (Optional)
- maxPriceLevel (Optional)

Libraries used by Google Maps Javascript API

- Canvaslayer
- Geolocationmarker
- Googleearth
- Infobubble
- Maplabel
- Markclusterer
- Richmaker
- Storelocator

Google map styling terms-

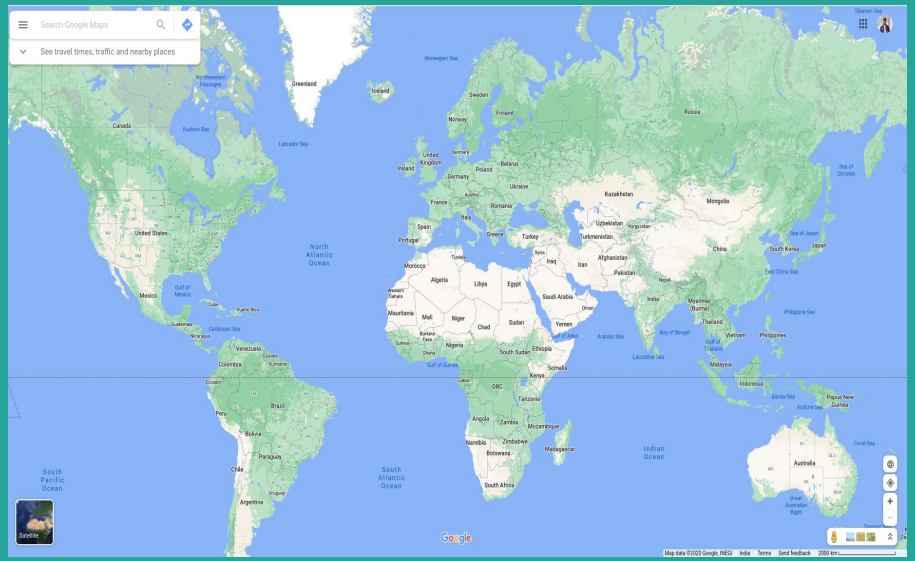
Marker

A marker identifies a location on a map. By default, a marker uses a standard image. Markers can display custom images, in which case they are usually referred to as "icons."

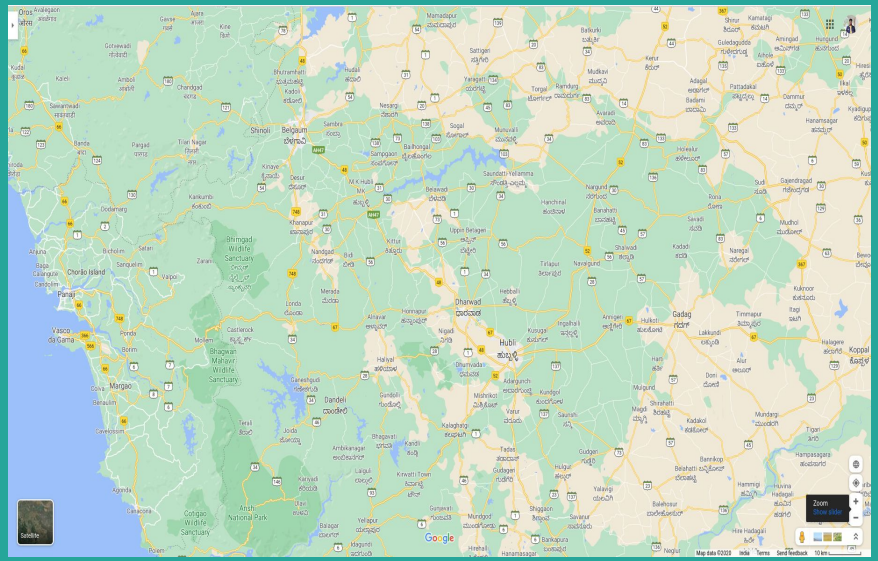
```
let marker;
marker = new google.maps.Marker({
  map, //map name to which it should be placed
  draggable: true,
  animation: google.maps.Animation.DROP,
  position: { lat: 59.327, lng: 18.067 }, //Initial position
});
```

Zooming

Level 0

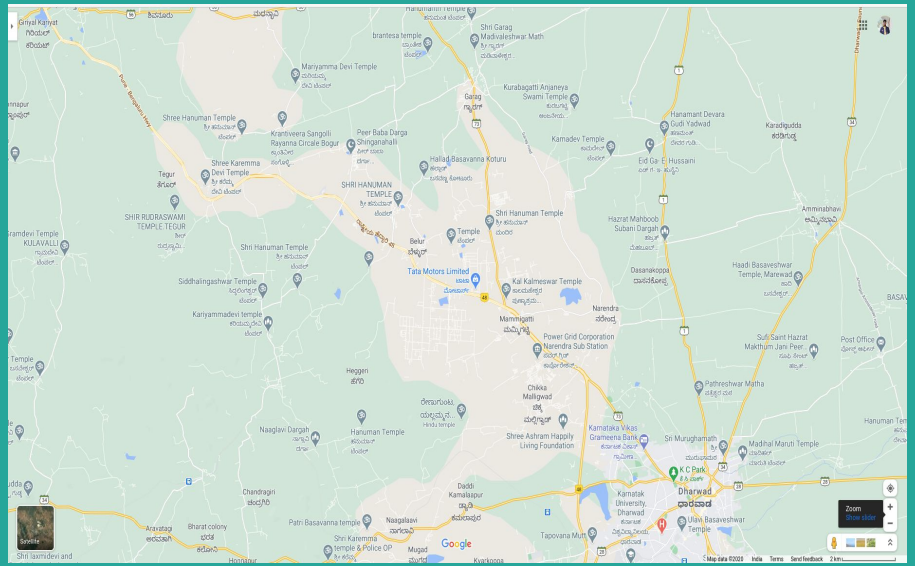


Level 7

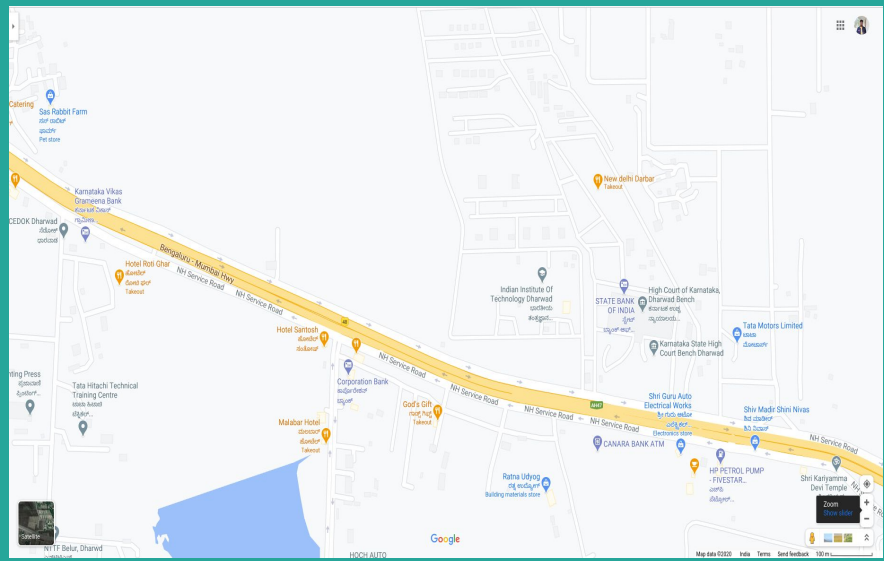


Zooming

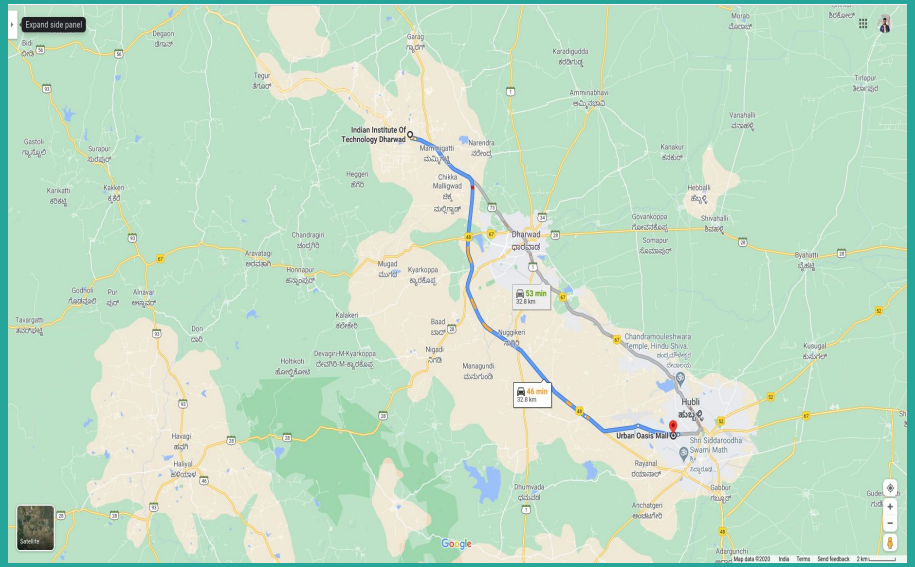
Level 10



Level 14



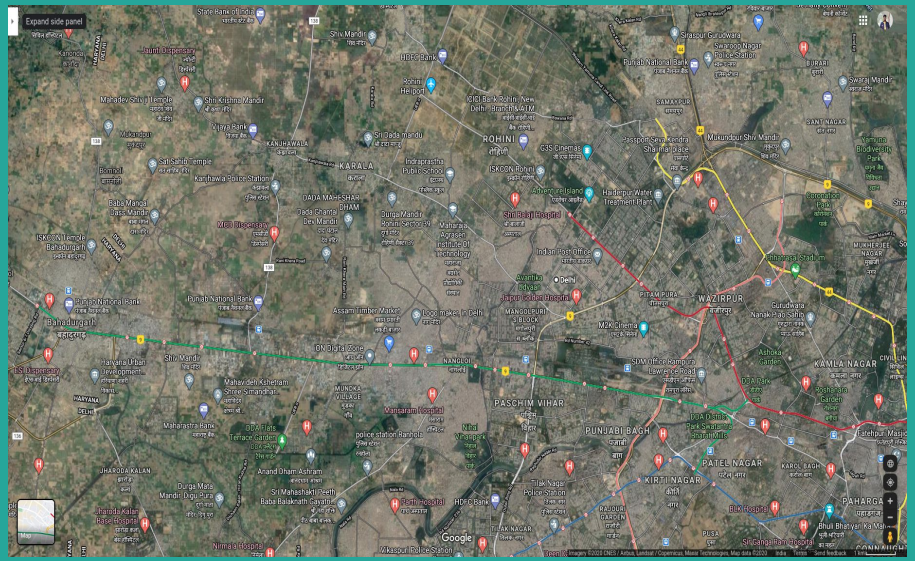
Direction



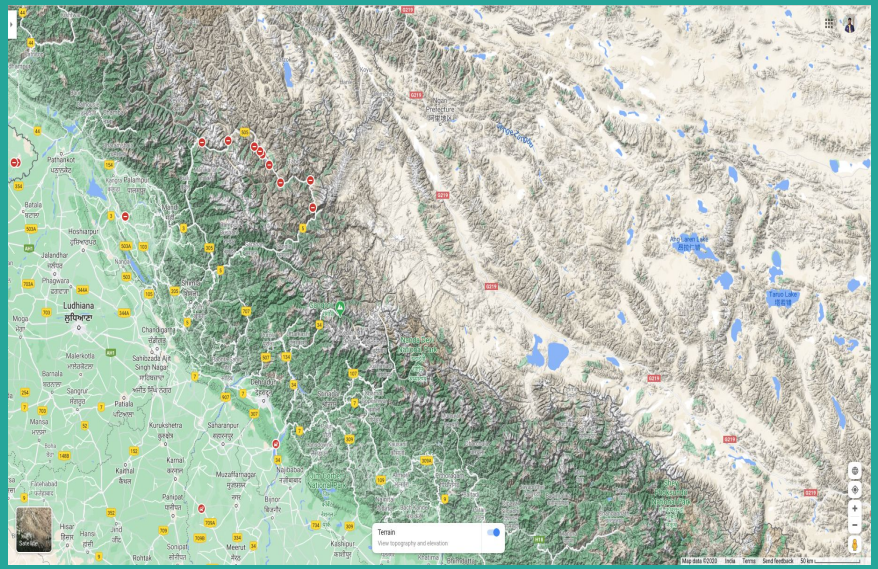
Different routes

Views

Satellite



Terrain



Services

Provided By Google Map

- Directions Service
- Distance Matrix Service
- Geocoding Service
- Embedding Api

Geocoding Service

- Converting human readable address to geographical coordinates and vice versa.

geocode() method of google.maps.geocoder constructor is used. It takes address or lat/lng or place id value as the input parameters and Results in an array which has several components such as lat/lng literals, formatted address, unique place id for that address and an address array which has parts of an address divided into categories such as sublocality ,locality, city, state, postal code etc.

CODE SNIPPET

```
const geocoder = new google.maps.Geocoder();
geocoder.geocode({ address: "IIT Dharwad"}, (results, status) => {
    if (status === "OK") {
        console.log(results)
    }
    else {
      alert("Geocode was not successful" + status);
    }
```

RESPONSE FORMAT

```
[{
  address_components: [{
  long_name: "High Court",
  short_name: "High Court",
  types: ["landmark"]
}, {
  long_name: "PB Road",
  short_name: "PB Road",
  types: ["route"]
},{..}],

formatted_address: "WALMI Campus, PB Road, near High Court,
            Anjaneya Nagar, Karnataka 580011, India",
  geometry: {
    location: { lat: 15.51, lng: 74.92 },
  },
place_id: "ChIJ30gAY3QzvzsRRhymN_RQis4"}
..]
```

Directions Service

Directions Service calculate directions(routes) for given methods of transportation and returns an efficient path between origin and destination.

Travel time is the primary factor which is optimized, but other factors such as distance, number of turns and many more may be taken into account.

Directions are displayed as a polyline drawing the route on a map or additionally as a series of textual descriptions such as ‘move 5 KM and then turn left ‘ etc..

Driving is default travel mode. Origin, Destination, travel mode are mandatory parameters whereas there are some other optional field parameters as well such as avoidHighways, avoidTolls etc.

For origin and destination field we can give any of among lat/lng literal or place id of that place or human readable address.

travelMode options - we can give Driving or Walking or Transit or Bicycling.

The other parameters can be tuned according to our requirements.

Directions service may return more than one possible itinerary as an array of separate routes each of which has detailed information. We will pass our response object to DirectionsRenderer object, which can automatically handle displaying the result on a map because given route information by google is not human readable and needs a renderer for that purpose.

Request can be made like as -

```
const directionsService = new google.maps.DirectionsService();
const directionsRenderer = new google.maps.DirectionsRenderer();
directionsService.route({
      origin: "Hubli",
      destination: "Dharwad",
      travelMode: google.maps.TravelMode.DRIVING,},
    (response, status) => {
     if (status === "OK") {
        console.log(response)
        directionsRenderer.setDirections(response);
      }
else {
        window.alert("Directions request failed due to " + status);
      }});
```

Distance Matrix Service

Distance Matrix service computes travel distance and journey duration between multiple origins and destinations using a given mode of transportation.

This service can be accessed by `google.maps.DistanceMatrixService` constructor object and we pass our parameters in `getDistanceMatrix()` method which will initiate a request to the service.

origins , destinations, travel mode are mandatory request parameters and there are some optional parameters as well. A callback function is also used to handle status code and the result object.

* This service does not return detailed route information. If user wants Route information, including polylines and textual directions, we can use Directions Service.

Origin and destination can be an array containing one or more address/ Latlng literal / place ID values.

CODE SNIPPET

```
const service = new google.maps.DistanceMatrixService();
  service.getDistanceMatrix(
    {
      origins: ["IIT Dharwad, karnataka"],
      destinations: ["Dharwad railway station, dharwad"],
      travelMode: google.maps.TravelMode.DRIVING,
      avoidHighways: false,
      avoidTolls: false,
    },
    (response, status) => {
      if (status !== "OK") {
        alert("Error was: " + status);
      } else {
        console.log(response)
      })
```

RESPONSE FORMAT

```
{
destinationAddresses: ["Dharwad, Malmaddi, Dharwad,..
                       Karnataka 580001, India"],
originAddresses: ["WALMI Campus, PB Road, near High..
        Court, Anjaneya Nagar, Karnataka 580011, India"],
rows: [{
  elements: [{
    distance: {
      text: "14.1 km",
      value: 14118 },
    duration: {
      text: "25 mins",
      value: 1519},
    status: "OK"
}] }]}
```

Map Embedding

Embedding a google map helps us to show some particular location in the map. This is a static google map. We can generally see map embedding in the ‘contact us’ page of a website/App. This can be done by a simple HTTP request. We can place google maps in the iframe tag on html pages.

Embed URL Format :

`https://www.google.com/maps/embed/v1/MODE?key=YOUR_API_KEY&parameters`

Mode can be place, search, view, directions, or street view.

The parameters include Mode - specific parameters as well as optional parameters.

REFERENCES

- https://developers.google.com/maps/documentation/javascript/overview

  (Main Google Maps API documentation)
- https://en.wikipedia.org/wiki/Google_Maps

Thank You ;-)