14. Google Maps API

충남대학교 컴퓨터공학과
데이타베이스시스템 연구실

Google Maps API Tutorial *Try it!*

- **What is Google Maps?**
  - Allows you to display maps on your web site

- **Google Maps API**
  - Google Maps API (**A**pplication **P**rogramming **I**nterface)
  - Let you customize maps and the information on maps

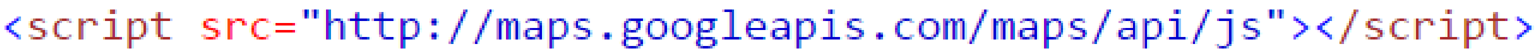
- ***Google Maps JavaScript API v3 (on Google Developers site)***

- ***Google Maps API reference (on www.w3schools.com)***

Google Maps Basic

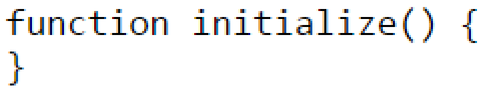
- Create a Basic Google Map *Try it!*

1. Load the Google API



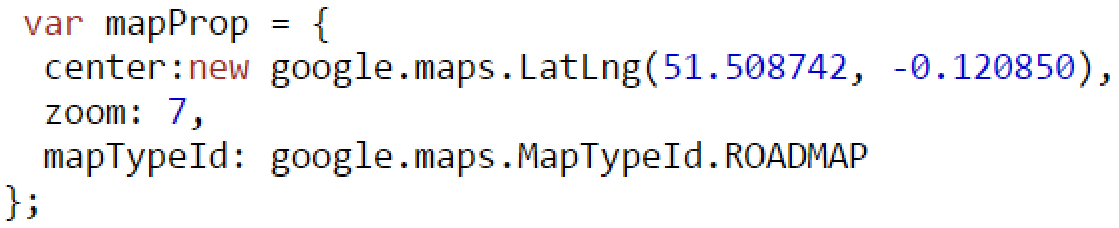
```
<script src="http://maps.googleapis.com/maps/api/js"></script>
```

2. Set Map Properties
   - create a function to initialize the map



```
function initialize() {
}
```

   - Create an object (mapProp) to define the properties for the map



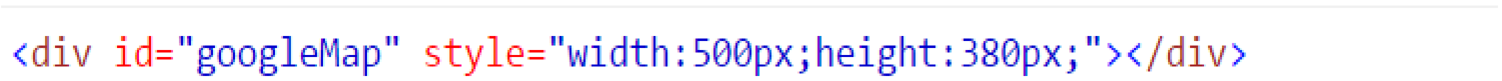
```
var mapProp = {
 center:new google.maps.LatLng(51.508742, -0.120850),
 zoom: 7,
 mapTypeId: google.maps.MapTypeId.ROADMAP
};
```

Google Maps Basic (cont'd)

- **center** property
  - Specifies where to center the map
  - Create a **LatLng** object to center the map on a specific point (latitude, longitude)
- **zoom** property
  - Specifies the zoom level for the map
  - Zoom:0 shows a map of the Earth fully zoomed out
  - Higher zoom levels zoom in at a higher resolution
- **mapTypeId** property
  - Specifies the map type to display
  - Map types
    - ROADMAP (normal, default 2D map)
    - SATELLITE (photographic map)
    - HYBRID (photographic map + roads and city names)
    - TERRAIN (map with mountains, rivers, etc.)

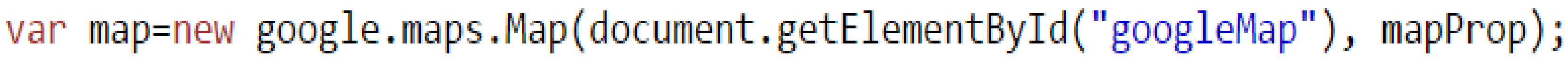
Google Maps Basic (cont'd)

3. Create a Map Container



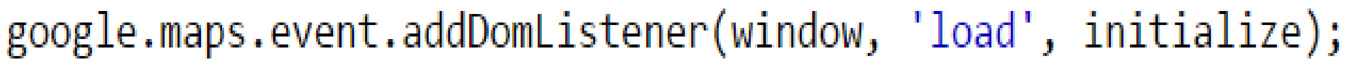
```
<div id="googleMap" style="width:500px;height:380px;"></div>
```

4. Create a Map Object



```
var map=new google.maps.Map(document.getElementById("googleMap"), mapProp);
```

5. Add an Event Listener to Load the Map



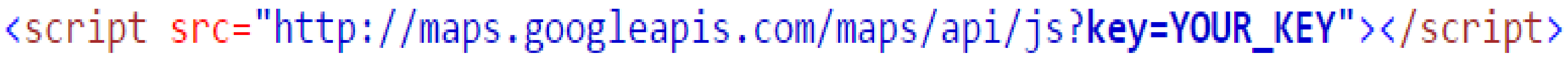
```
google.maps.event.addDomListener(window, 'load', initialize);
```

Google Maps Basic (cont'd)

- **Asynchronously Loading** *Try it!*
  - It is possible to load the Google Maps API on demand.
  - This example uses window.onload to load the Google Maps API after the page has fully loaded.

- **Multiple Maps** *Try it!*

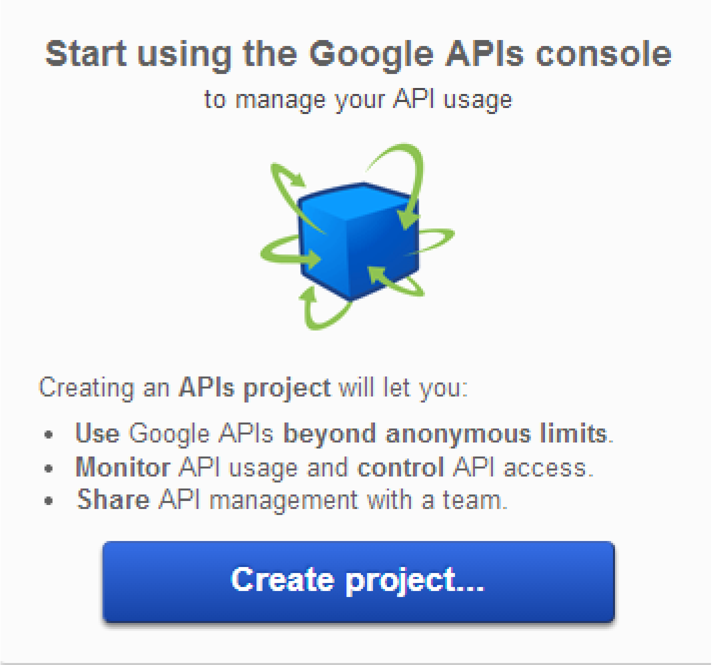
- **Google API Key**
  - If you plan for heavier traffic, you should get a free API key from Google
  - Go to https://console.developers.google.com to get a free key.



```
<script src="http://maps.googleapis.com/maps/api/js?key=YOUR_KEY"></script>
```

참고 : Google Maps API Key

- **How to get started?**
  - Need a specific API Key from Google
  - Key is free
  - Go to https://code.google.com/apis/console/ and log in with your Google Account



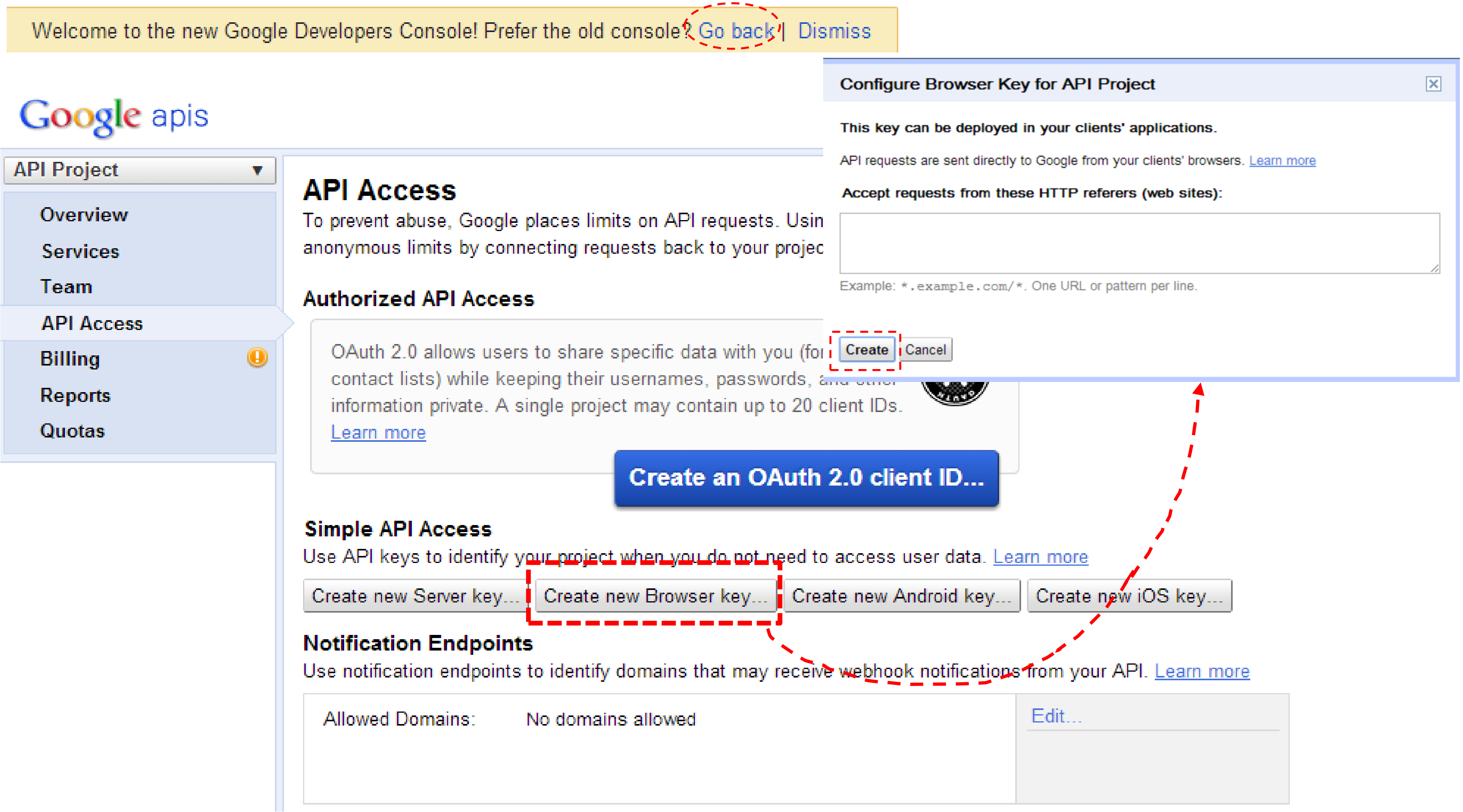
Start using the Google APIs console

to manage your API usage

Creating an **APIs project** will let you:

- **Use** Google APIs **beyond anonymous limits**.
- **Monitor** API usage and **control** API access.
- **Share** API management with a team.

To get an API key



Welcome to the new Google Developers Console! Prefer the old console? Go back | Dismiss

Google apis

API Project ▼

- Overview
- Services
- Team
- API Access
- Billing
- Reports
- Quotas

API Access

To prevent abuse, Google places limits on API requests. Usin
anonymous limits by connecting requests back to your projec

Authorized API Access

OAuth 2.0 allows users to share specific data with you (fo
contact lists) while keeping their usernames, passwords, and other
information private. A single project may contain up to 20 client IDs.
Learn more

Create an OAuth 2.0 client ID...

Simple API Access

Use API keys to identify your project when you do not need to access user data. Learn more

Create new Server key... | Create new Browser key... | Create new Android key... | Create new iOS key...

Notification Endpoints

Use notification endpoints to identify domains that may receive webhook notifications from your API. Learn more

Allowed Domains: No domains allowed

Edit...

Configure Browser Key for API Project

This key can be deployed in your clients' applications.

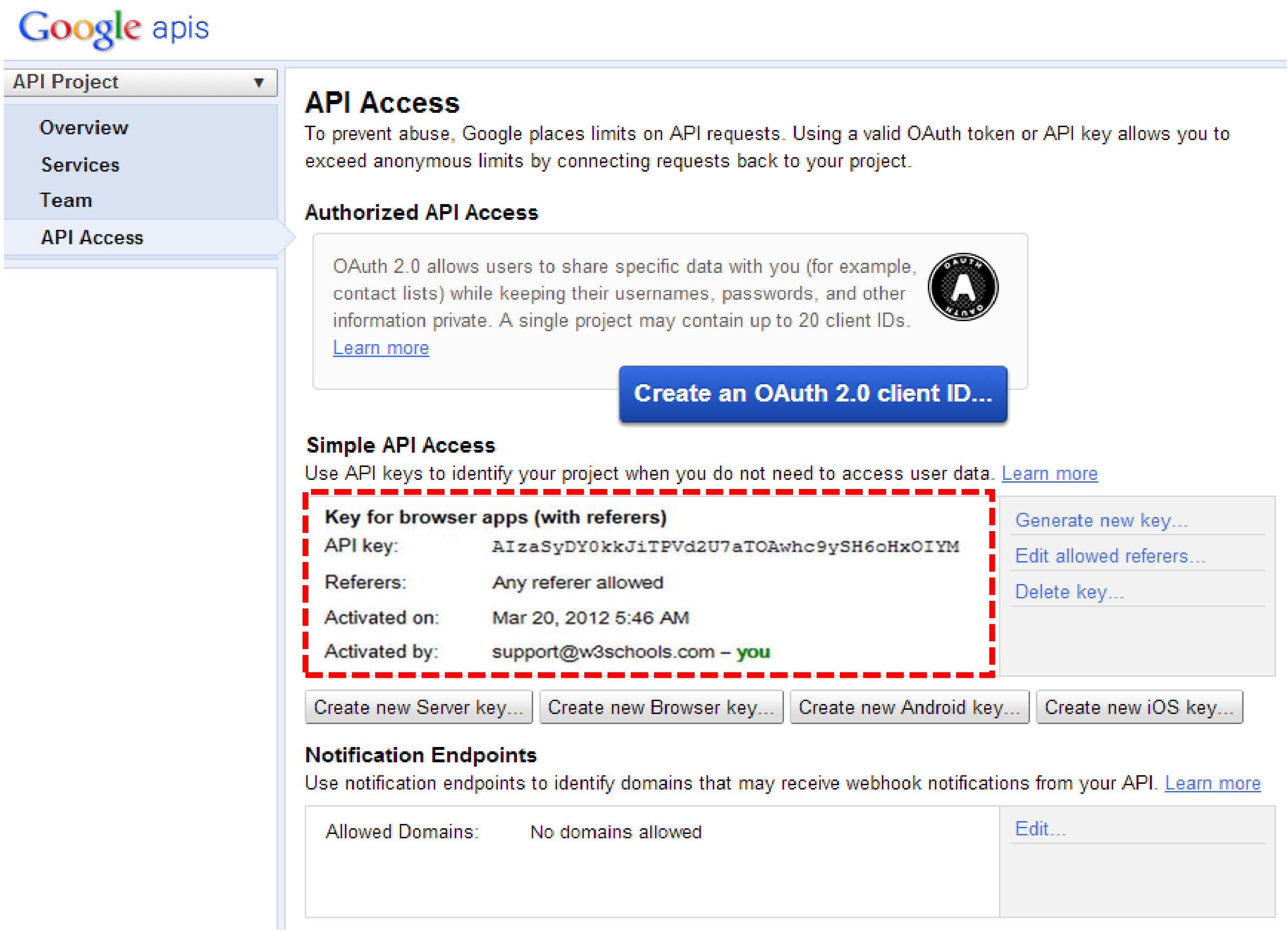
API requests are sent directly to Google from your clients' browsers. Learn more

Accept requests from these HTTP referers (web sites):

Example: `*.example.com/*`. One URL or pattern per line.

Create | Cancel

To get an API key (cont'd)



API Access

To prevent abuse, Google places limits on API requests. Using a valid OAuth token or API key allows you to exceed anonymous limits by connecting requests back to your project.

Authorized API Access

OAuth 2.0 allows users to share specific data with you (for example, contact lists) while keeping their usernames, passwords, and other information private. A single project may contain up to 20 client IDs. Learn more

Simple API Access

Use API keys to identify your project when you do not need to access user data. Learn more

Key for browser apps (with referers)

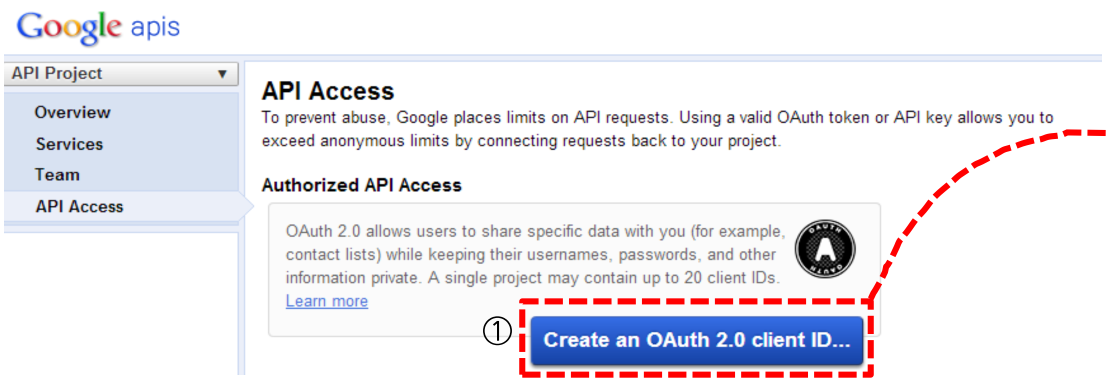
API key:	AIzaSyDY0kkJiTPVd2U7aTOAwhc9ySH6oHxOIYM
Referers:	Any referer allowed
Activated on:	Mar 20, 2012 5:46 AM
Activated by:	support@w3schools.com – you

Notification Endpoints

Use notification endpoints to identify domains that may receive webhook notifications from your API. Learn more

Allowed Domains: No domains allowed

To create Client ID



Google apis

API Project ▾

- Overview
- Services
- Team
- API Access

API Access

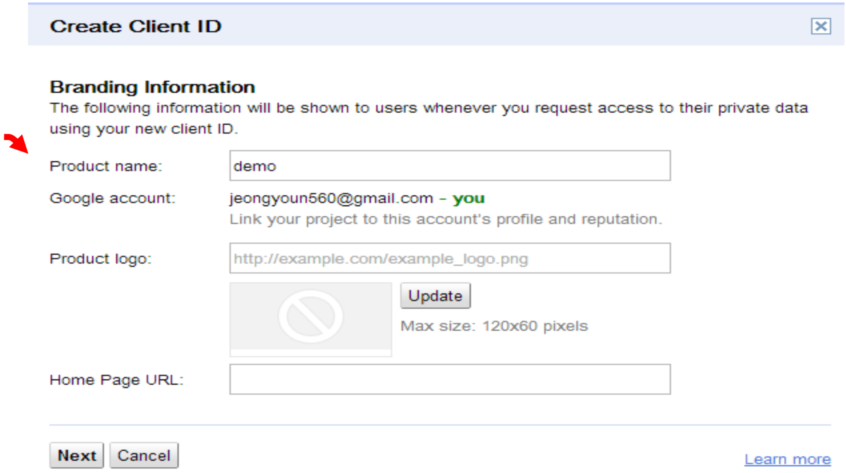
To prevent abuse, Google places limits on API requests. Using a valid OAuth token or API key allows you to exceed anonymous limits by connecting requests back to your project.

Authorized API Access

OAuth 2.0 allows users to share specific data with you (for example, contact lists) while keeping their usernames, passwords, and other information private. A single project may contain up to 20 client IDs. Learn more

① Create an OAuth 2.0 client ID...

②



Create Client ID

Branding Information

The following information will be shown to users whenever you request access to their private data using your new client ID.

Product name: demo

Google account: jeongyoun560@gmail.com – you

Link your project to this account's profile and reputation.

Product logo: http://example.com/example_logo.png

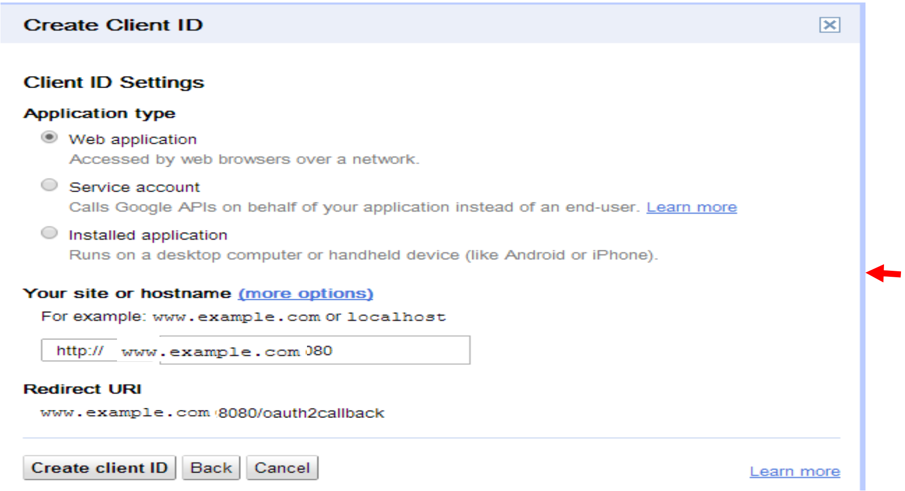
Update

Max size: 120x60 pixels

Home Page URL:

Next | Cancel | Learn more

③



Create Client ID

Client ID Settings

Application type

● Web application
Accessed by web browsers over a network.

○ Service account
Calls Google APIs on behalf of your application instead of an end-user. Learn more

○ Installed application
Runs on a desktop computer or handheld device (like Android or iPhone).

Your site or hostname (more options)

For example: www.example.com or localhost

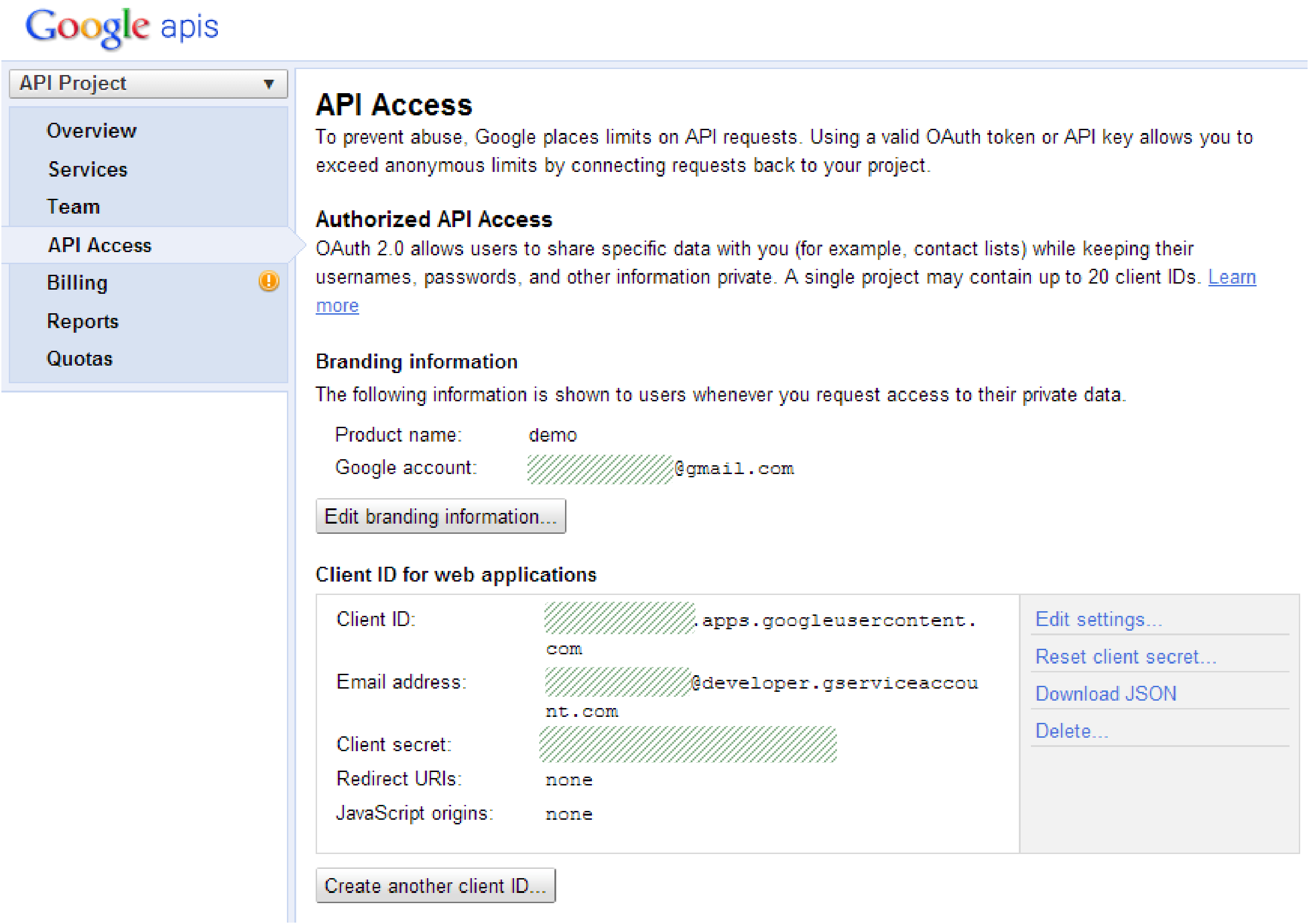
http:// www.example.com)80

Redirect URI

www.example.com 8080/oauth2callback

Create client ID | Back | Cancel | Learn more

To create Client ID (cont'd)



API Access

To prevent abuse, Google places limits on API requests. Using a valid OAuth token or API key allows you to exceed anonymous limits by connecting requests back to your project.

Authorized API Access

OAuth 2.0 allows users to share specific data with you (for example, contact lists) while keeping their usernames, passwords, and other information private. A single project may contain up to 20 client IDs. Learn more

Branding information

The following information is shown to users whenever you request access to their private data.

Product name:	demo
Google account:	[redacted]@gmail.com

Edit branding information...

Client ID for web applications

Client ID:	[redacted].apps.googleusercontent.com
Email address:	[redacted]@developer.gserviceaccount.com
Client secret:	[redacted]
Redirect URIs:	none
JavaScript origins:	none

Create another client ID...

Google Maps Overlays *Try it!*

- **Google Maps - Overlays**
  - Objects on the map that are bound to latitude/longitude coordinates
  - Several types of overlays
    - Marker - Single locations on a map. Markers can also display custom icon images
    - Polyline - Series of straight lines on a map
    - Polygon - Series of straight lines on a map, and the shape is "closed"
    - Circle and Rectangle
    - Info Windows - Displays content within a popup balloon on top of a map
    - Custom overlays

Google Maps Overlays (cont'd)

- **Add a Marker** *Try it!*
  - The marker constructor creates a marker
  - Note that position property must be set for the marker to display
  - Add the marker to the map by using the setMap( ) method.

- **Animate the Marker** *Try it!*
  - Animation property
    - which animation to play when marker is added to a map
    - Constant : BOUNCE, DROP

- **Icon Instead of Marker** *Try it!*
  - Specifies an image(icon) to use instead of the default marker

Google Maps Overlays (cont'd)

- **Polyline** *Try it!*
  - A line that is drawn through a series of coordinates in an ordered sequence
  - Properties
    - path : specifies several latitude/longitude coordinates for the line
    - strokeColor : specifies a hexadecimal color for the line (format :"#FFFFFF")
    - strokeOpacity : specifies the opacity of the line (a value between 0.0 and 1.0)
    - strokeWeight : Specifies the weight of the line's stroke in pixels
    - editable : defines whether the line is editable by users (true/false)

Google Maps Overlays (cont'd)

- **Polygon** *Try it!*
  - Is similar to a Polyline in that it consists of a series of coordinates in an ordered sequence.
  - However, polygons are designed to define regions within a closed loop
  - Properties
    - path : specifies several LatLng coordinates for the line (first and last coordinate are equal)
    - strokeColor : specifies a hexadecimal color for the line (format :"FFFFFF")
    - strokeOpacity : specifies the opacity of the line (a value between 0.0 and 1.0)
    - strokeWeight : specifies the weight of the line's stroke in pixels
    - fillColor : specifies a hexadecimal color for the area within the enclosed region (format :"#FFFFFF")
    - fillOpacity : specifies the opacity of the fill color (a value between 0.0 and 1.0)
    - editable : defines whether the line is editable by users (true/false)

Google Maps Overlays (cont'd)

- Circle *Try it!*
  - Properties
    - center : specifies the goole.maps.LatLng of the center of the circle
    - radius : specifies the radius of the circle, in meters
    - strokeColor : specifies a hexadecimal color for the line around the circle(format :"#FFFFFF")
    - strokeOpacity : specifies the opacity of the stroke color (a value between 0.0 and 1.0)
    - strokeWeight : specifies the weight of the line's stroke in pixels
    - fillColor : specifies a hexadecimal color for the area within the circle(format :"FFFFFF")
    - fillOpacity : specifies the opacity of the fill color (a value between 0.0 and 1.0)
    - editable : defines whether the circle is editable by users (true/false)

Google Maps Overlays (cont'd)

- InfoWindow   Try it!
    - Show an InfoWindow with some text content for a marker
        - open(map, anchor)
            - Opens this InfoWindow on the given map.
            - An infoWindow can be associated with an anchor

Google Maps Events *Try it!*

- **Click The Marker to Zoom** *Try it!*

- **Pan Back to Marker** *Try it!*
  - Add an event handler to the map for changes to the 'center' property and pan the map back to the marker after 3 seconds on a center_changed event
  - panTo(latLang:LatLan)
    - Changes the center of the map to the given LatLng

- **Open an InfoWindow When Clicking on The Marker** *Try it!*
  - Click on the marker to show an infoWindow the some text

Google Maps Events (cont'd)

- **Set Markers and Open InfoWindow for Each Marker** ***Try it!***
  - placeMarker() function
    - Places a marker where the user has clicked , and show an infowindow with the latitudes and longitudes of the marker

Google Maps Controls *Try it!*

The Default Controls

- Zoom : displays a slider or "+/-" buttons to control the zoom level of the map
- Pan : displays a pan control for panning the map
- MapType : lets the user toggle between map types (roadmap and satellite)
- Street View :displays a Pegman icon which can be dragged to the map to enable Street View

More Controls

- Scale : displays a map scale element
- Rotate : displays a small circular icon which allows you to rotate maps
- Overview Map : display a thumbnail overview map reflecting the current map viewport within a wider area
- You can specify which controls to show when creating the map (inside MapOptions) or by calling setOptions ( ) to change the map's options.

Google Maps Controls (cont'd)

- **Disabling The Default Controls** *Try it!*
  - Set the Map's disable DefaultUI property to true

- **Turn On All Controls** *Try it!*
  - Some controls appear on the map by default; while others will not appear unless you set them
  - Adding or removing controls from the map is specified in the Map options object
  - Set the control to true to make it visible
  - Set the control to false to hide it

Google Maps Controls (cont'd)

Modifying Controls

- The controls can be modified by specifying control options fields.
- For example 1   *Try it!*
  - Options for modifying a Zoom control are specified in the zoomControlOptions field.
  - google.maps.ZoomControlSytle.SMALL : displays a mini-zoom control(only + and-buttons)
  - google.maps.ZoomControl.LARGE : displays the standard zoom slider control
  - google.maps.ZoomContorl.DEFAULT:picks the best zoom control based on device and map size

Google Maps Controls (cont'd)

- For example 2 *Try it!*
  - Options for modifying a control are specified in the mapTypeControlOptions field.
  - google.maps.MapTypeControlStyle.HORIZONTAL_BAR : display one button for each map type
  - google.maps.MapTypeControlStyle.DROPDOWN_MENU : select map type via a dropdown menu
  - google.maps.MapTypeControlStyle.DEFAULT : displays the "default" behavior (depends on screen size)

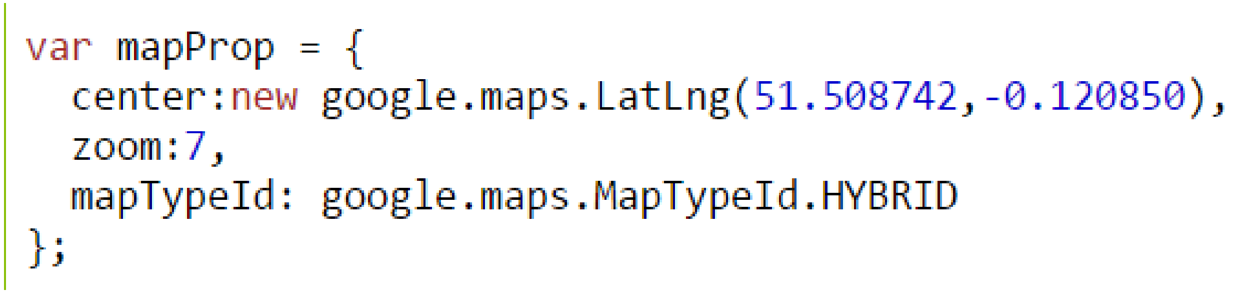
Custom Controls *Try it!*

- Create a custom control that always takes you back to London, when clicked

Google Maps Types *Try it!*

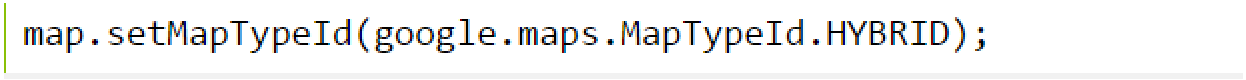
Basic Map Types

- ROADMAP (normal, default 2D map)
- SATELLITE (photographic map)
- HYBRID (photographic map + roads and city names)
- TERRAIN (map with mountains, rivers, etc)
- Specified either within the Map properties object , with the mapTypeId property



```
var mapProp = {
  center:new google.maps.LatLng(51.508742,-0.120850),
  zoom:7,
  mapTypeId: google.maps.MapTypeId.HYBRID
};
```

- Or by calling the map's setMapTypeId() method



```
map.setMapTypeId(google.maps.MapTypeId.HYBRID);
```

Google Maps Types (cont'd)

45° Perspective View *Try it!*

- The map types SATELLITE and HYBRID support a 45° perspective imagery view for certain locations (only at high zoom levels)
- If you zoom into a location with 45° imagery view, the map will automatically alter the perspective view.
- In addition
  - A compass wheel around the Pan control, allowing you to rotate the image
  - A Rotate control between the Pan and Zoom controls, allowing you to rotate the image 90°
  - A toggle control for displaying the 45° perspective view, under Satellite control/label

Disable 45° Perspective View – setTilt(0) *Try it!*

- You can disable 45° perspective view by callying setTilt(0) on the Map object