# Let’s use Google Maps API as an Example

- This app will help users explore "hidden gems" around Seattle, WA, showcasing how to integrate and interact with Google Maps within an app.

# Let’s use Google Maps API as an Example

Step 1: Set Up Your Project

- Create a New Android Project:
  - Open Android Studio and start a new project.
  - Choose ”Empty Views Activity" for simplicity.
- Add Google Maps Dependencies:
  - Open your build.gradle (Module: app) file.
  - Add the following line to include the Google Maps API dependency:
  - implementation 'com.google.android.gms:play-services-maps:17.0.1'

# Let’s use Google Maps API as an Example

## Step 2: Add Permissions and Generate API Key

- Add Internet Permission:
  - In your AndroidManifest.xml, add the following permission:

    -
      ```
      <uses-permission android:name="android.permission.INTERNET"/>
      ```
- Get a Google Maps API Key:
  - Visit the Google Cloud Console, create a new project, and enable the Google Maps API.
  - Generate an API key which will be used to authenticate your application with Google services.
  - Add the API key to your AndroidManifest.xml:

    -
      ```
      <meta-data
          android:name="com.google.android.geo.API_KEY”
          android:value="YOUR_API_KEY"/>
      ```

# Let’s use Google Maps API as an Example

## Step 3: Integrate Google Maps

- Modify the layout XML file (activity_main.xml):
  - Include a Fragment that will display the map:
  - 
    ```
    <fragment
      android:id="@+id/map”
      android:name="com.google.android.gms.maps.SupportMapFragment”
      android:layout_width="match_parent”
      android:layout_height="match_parent"/>
    ```
  - This XML fragment embeds a map into the activity layout.

# Let’s use Google Maps API as an Example

Step 3: Integrate Google Maps

- Initialize the map in your Activity (MainActivity.kt):
  - val mapFragment = supportFragmentManager
  -     .findFragmentById(R.id.map) as SupportMapFragment
  - mapFragment.getMapAsync { googleMap ->
  -     // This is where we use the Google Maps API
  -     setUpMap(googleMap)
  - }

# Let’s use Google Maps API as an Example

## Step 4: Customize the Map for Seattle, WA

- Configure Map Settings (MainActivity.kt):

```
fun setUpMap(map: GoogleMap) {
    map.apply {
        mapType = GoogleMap.MAP_TYPE_NORMAL
        uiSettings.isZoomControlsEnabled = true
        moveCamera(CameraUpdateFactory.newLatLngZoom(LatLng(47.6062, -122.3321), 12f))
    }
}
```

# Let’s use Google Maps API as an Example

## Step 5: Add Markers for Hidden Gems

- Add Markers on the Map (MainActivity.kt):
  - Define a list of coordinates representing hidden gems and add markers:
    - val hiddenGems = listOf(
    - LatLng(47.6101, -122.3421), // Sculpture Park
    - LatLng(47.6289, -122.3426)  // Gas Works Park
    - )
    - hiddenGems.forEach { location ->
    - map.addMarker(MarkerOptions().position(location).title("Hidden Gem"))
    - }

# Let’s use Google Maps API as an Example

## Step 6: Interactivity and Testing

- Handle Marker Click Events:
  - Provide more details when a marker is tapped:
    - map.setOnMarkerClickListener { marker ->
    - // Code to display an info window or another activity with details
    - false
    - }