

City Tourism Guide: Project Configuration and Google Maps API Integration

Introduction

The City Tourism Guide is an Android mobile application that enhances a user's ability to explore tourist attractions within a city such as Toronto. The app incorporates a map interface to show the user's location, static tourist spots, and dynamic nearby places using Google services. This report walks through the essential process of acquiring an API key from Google Cloud and setting up the project in Android Studio to ensure it runs correctly.

Section 1: Getting a Google Maps API Key

Google Maps and Places APIs require an API key for access. This key is used to authenticate your app when communicating with Google's servers. It must be created through the Google Cloud Console and properly restricted for security.

Step 1: Visit the Google Cloud Console

Start by navigating to <https://console.cloud.google.com>. Sign in with your Google account if prompted.

Step 2: Set Up a Project

In the dashboard, click the dropdown at the top and choose "New Project." Provide a name that is easy to recognize (e.g., "TourismApp2025"). Once you create it, this becomes the active project for which settings and services will be applied.

Step 3: Enable Required APIs

From the sidebar menu, go to "APIs & Services" > "Library." Use the search function to find and enable:

Maps SDK for Android – displays the map

Places API – provides access to location data such as restaurants, landmarks, etc.

Enabling these services allows the app to render maps and fetch dynamic places nearby the user.

Step 4: Create and Manage API Key

Next, go to "Credentials" under "APIs & Services" and select "Create Credentials" > "API Key." A unique key will be generated instantly.

Step 5: Restrict the API Key

To avoid unauthorized usage:

- Under Application Restrictions, choose "Android apps."
- Provide your app's package name and its SHA-1 fingerprint (which can be found using Android Studio tools).
- Under API Restrictions, limit access to only the Maps SDK for Android and Places API.

After setting restrictions, save the configuration. You'll need this key for integrating it into the app later.

Section 2: Running the Project in Android Studio

Now that the API key is ready, you can move forward with setting up and executing the application in Android Studio.

Step 1: Start a New Android Project

Open Android Studio and select "New Project." Choose "Empty Activity," then:

- Set Kotlin as the programming language
- Choose a minimum SDK of API 24 (Android 7.0) or above
- Name the project "CityTourismGuide"

This will generate a basic project structure including `MainActivity`.

Step 2: Prepare Project Files

You will need at least two layout files:

- One for the map screen with a MapView and filters
- One for the place details screen where users see information and navigation options

Along with the layout files, make sure you've structured the app to handle location access, map rendering, and UI interactions properly.

Step 3: Connect Required Permissions

For the app to function, it must be allowed to:

- Access the device's location (to show current position)
- Access the internet (to retrieve map and place data)

These permissions must be declared, and location permission must be requested at runtime when the app starts.

Step 4: Use Version Catalogs for Dependencies

To keep the project well-organized and up-to-date, dependencies like Google Maps, Places API, Location Services, and Volley should be declared using version catalogs. This makes it easier to manage library versions and ensures better compatibility over time.

Step 5: Implement App Features

Once dependencies and permissions are in place, build out the core functionality:

- Displaying the map and current location
- Adding static markers for well-known tourist attractions
- Dynamically fetching places based on selected categories (e.g., parks, cafes)
- Showing place information on a new screen
- Launching Google Maps with directions when requested

The app uses a Spinner to filter categories and displays places based on user selection, which are then shown using markers on the map.

Section 3: Final Testing and Tips

Testing the App

Use both an emulator (with location enabled) and a real Android device to test the application. Make sure:

- The map loads and centers on the user's location
- The Spinner works to filter nearby places
- Tapping on markers displays detailed place info
- Navigation opens in Google Maps properly

Important Guidelines

- Do not expose your API key publicly.** Use Git ignore rules to avoid pushing it to public repositories.
- Billing must be enabled on your Google Cloud account**, even if usage is within the free tier.
- Handle errors properly**—especially those related to network requests or missing location permissions.
- Design the UI for clarity and simplicity**, making it easy for users to explore and navigate through places.

Conclusion:

By following this guide, users can successfully acquire a Google Maps API key, restrict it for safe usage, and configure an Android Studio project to build a functional and interactive tourism guide app. This project serves as a valuable learning tool for implementing real-time maps, dynamic data fetching, and mobile UI design using modern Android development practices. Whether testing on an emulator or real device, the key to success lies in careful setup, secure API usage, and thoughtful interface design.