Using Google Maps

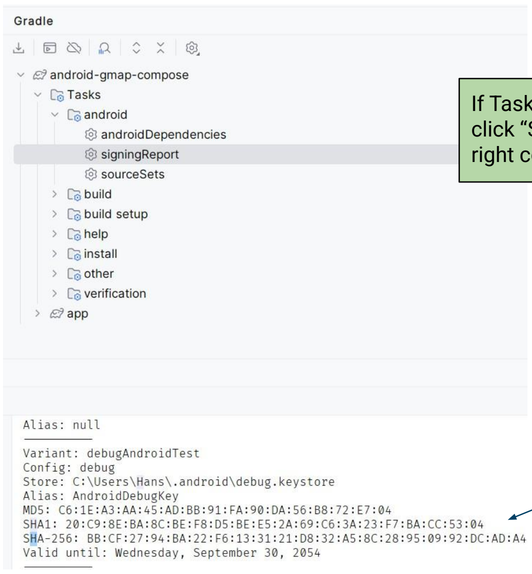
Step 0: Create Android Studio Project

- Select Empty Activity (Jetpack Compose)
- Add Google Maps Compose Dependency

```
 dependencies {
    // Android Maps Compose composables for the Maps SDK for Android
    // Warning: This library seems to require Kotlin 2.2.x
    implementation("com.google.maps.android:maps-compose:6.12.0")
}
```

- Open Gradle Panel
  - Tasks ⇒ Android ⇒ Signing Report
  - Copy SHA1 Fingerprint (next slide)

Step 1a: SHA-1 Certificate Fingerprint



```
Alias: null
───────────
Variant: debugAndroidTest
Config: debug
Store: C:\Users\Hans\.android\debug.keystore
Alias: AndroidDebugKey
MD5: C6:1E:A3:AA:45:AD:BB:91:FA:90:DA:56:B8:72:E7:04
SHA1: 20:C9:8E:BA:8C:BE:F8:D5:BE:E5:2A:69:C6:3A:23:F7:BA:CC:53:04
SHA-256: BB:CF:27:94:BA:22:F6:13:31:21:D8:32:A5:8C:28:95:09:92:DC:AD:A4
Valid until: Wednesday, September 30, 2054
───────────
```

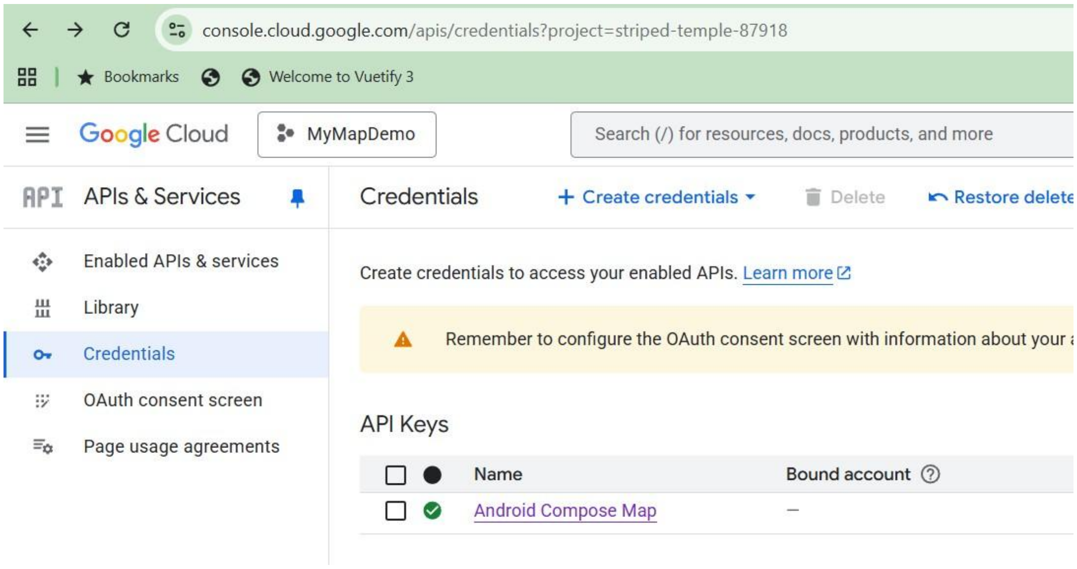
If Tasks ⇒ android ⇒ signingReport is not listed, click “Sync Project with Gradle File” at the upper right corner (or **Shift-Cmd-O**)

Copy SHA1

Step 1b: Create a Google Cloud Project

- https://cloud.google.com (or https://console.cloud.google.com)
- Create a new project
- (Enable Payment)
- Enable Maps SDK for Android (or iOS)
  - Generate & Copy the API key
  - **Restriction type**: Android Apps
  - **Package name**: copy from your Android Studio project

Generate API Key



console.cloud.google.com/apis/credentials?project=striped-temple-87918

Bookmarks Welcome to Vuetify 3

Google Cloud MyMapDemo Search (/) for resources, docs, products, and more

APIs & Services Credentials + Create credentials Delete Restore delete

- Enabled APIs & services
- Library
- Credentials
- OAuth consent screen
- Page usage agreements

Create credentials to access your enabled APIs. Learn more

Remember to configure the OAuth consent screen with information about your a

API Keys

		Name	Bound account
☐	●	Android Compose Map	—

Step 2a: Include API key In AndroidManifest

```
<application ....>
   <meta-data android:name="com.google.android.geo.API_KEY"
     android:value="AIza____________________” />
   <activity android:name=”.MainActivity”




   />
</application>
```

Concealing your API Key

- Add Secrets Gradle Plugin (on GitHub)
    - Add classpath in top-level build.gradle.kts

```
buildscript {
    dependencies {
        classpath(group = "com.google.android.libraries.mapsplatform.secrets-gradle-plugin”,
                 name = “secrets-gradle-plugin”,
                 version = “2.0.1")
   }
}
```

    - Add plugin in module build.gradle.kts

```
plugin {
    id("com.google.android.libraries.mapsplatform.secrets-gradle-plugin")
}
```

Concealing Your API Key

- Add the secrets block in build.gradle.kts

```
secrets {
   propertiesFileName = "secrets.properties"
   defaultPropertiesFileName = "local.defaults.properties"
}
```

- Create the two files secrets.properties and local.defaults.properties

```
# In secrets.properties (this file should be git-ignored)
MAPS_API_KEY=AIza______________________
```

```
# In local.defaults.properties
MAPS_API_KEY=DEFAULT_API_KEY
```

Replace Hardcoded API Key

```
<application ....>
   <meta-data android:name="com.google.android.geo.API_KEY"
     android:value="AIza____________________” />
   <activity android:name=”.MainActivity”
   />
</application>
```

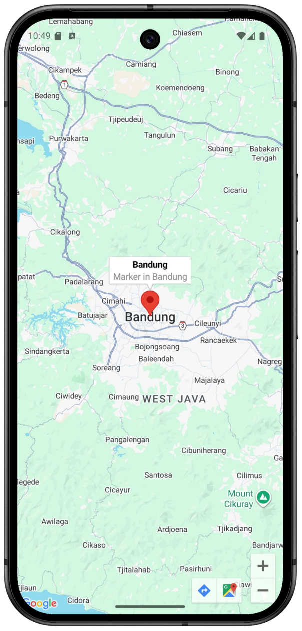
```
<application ....>
   <meta-data android:name="com.google.android.geo.API_KEY"
     android:value="${MAPS_API_KEY}” />
   <activity android:name=”.MainActivity”
   />
</application>
```

Show Google Maps

```
val bandung = LatLng(-6.9175, 107.6191)
val cameraPositionState = rememberCameraPositionState  {
    position = CameraPosition.fromLatLngZoom(bandung, 10f)
}
val cityMarker = rememberUpdatedMarkerState(position = bandung)
GoogleMap(modifier = Modifier. fillMaxSize(),
   cameraPositionState = cameraPositionState)  {

    Marker(state = cityMarker, title = “Bandung”,
           snippet = “Marker in Bandung”)

}
```


Bandung
Marker in Bandung
Bandung
WEST JAVA
Cimahi
Padalarang
Soreang
Purwakarta
Subang