< CC Beacon Library _ Quick Guide>

1. 라이브러리 적용

1-1. 안드로이드 스튜디오 (Android Studio)

1) Project 폴더 내에 libs 폴더를 생성합니다.

2) libs 폴더 내에 'cc-beacon-library-1.0.0.aar' 파일을 추가합니다.

3) 앱의 build.gradle 의 repositories에 flatDir을 추가하고

```
repositories {
    mavenCentral()
    flatDir {
      dirs 'libs'
    }
  }
```

dependencies에 compile ':cc-beacon-library-1.0.0:@aar'를 추가합니다.

```
dependencies {
    compile ':cc-beacon-library-1.0.0:@aar'
}
```

4) AndroidManifest.xml 의 application에 'ccbeaconService'를 등록하고 service를 이용하고자 하는 activity의 모드를 singleInstance로 설정합니다.

```
<application
        android:name="biz.ciycat.service.ccbeaconService"
        android:allowBackup="true"
        android:icon="@mipmap/cc_beacon_icon"
        android:label="@string/app_name"
        android:theme="@style/AppTheme" >
        <activity
            android:name=".MainActivity"
            android:label="@string/app_name"
            android:launchMode="singleInstance">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />

                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
    </application>
</manifest>
```

1-2. 이클립스 (Eclipse)

1) Project 의 libs 폴더 내에 'cc-beacon-library-1.0.0.jar' 파일을 추가합니다.

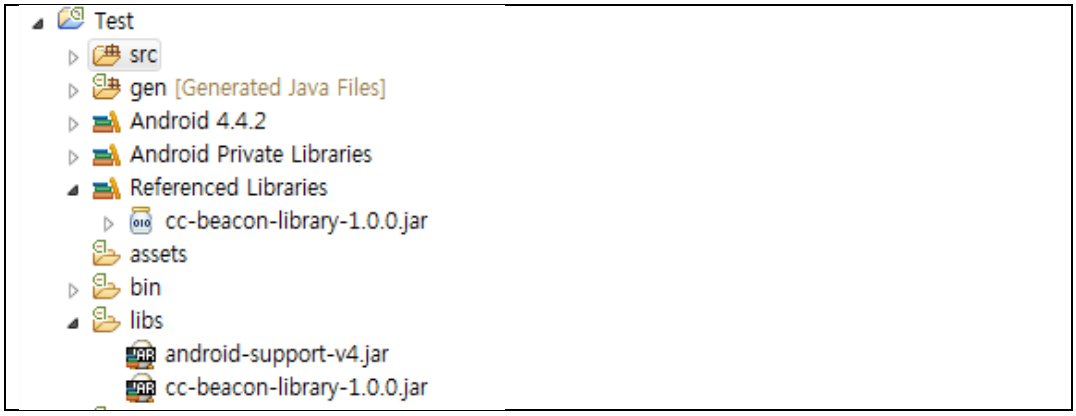
2) Project -> Properties -> Java Build Path 의 상단 탭 Libraries -> Add Library -> User Library -> User Libraries 를 선택합니다.

3) New 버튼을 눌러 User library name에는 원하는 name 을 적습니다.

4) User libraries가 만들어지면 Add JARs...를 선택하시고 처음에 libs 폴더에 넣었던 jar파일을 선택합니다.

5) Project의 properties 의 Java Build Path 의 상단 탭 Order and Export 에서 3)에 선택한 폴더를 check합니다.

6) 다음과 같이 library 파일이 추가된 모습을 확인할 수 있습니다.



2. 블루투스 퍼미션 추가

AndroidMainfest.xml 파일에 Permission을 추가합니다.

```
<uses-permission android:name="android.permission.BLUETOOTH"/>
<uses-permission android:name="android.permission.BLUETOOTH_ADMIN"/>
```

3. 샘플 코드

1) Monitoring 예제 코드

```
public class MonitoringActivity extends Activity implements BeaconConsumer {
        protected static final String TAG = "MonitoringActivity";
        private BeaconManager beaconManager =
    BeaconManager.getInstanceForApplication(this);
```

```
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_monitoring);
       beaconManager.bind(this);
     }
    @Override
    protected void onDestroy() {
        super.onDestroy();
        beaconManager.unbind(this);
    }
    @Override
    public void onBeaconServiceConnect() {
        beaconManager.setMonitorNotifier(new MonitorNotifier() {
        @Override
        public void didEnterRegion(Region region) {
          Log.i(TAG, "I just saw an beacon for the first time!");
        }

        @Override
        public void didExitRegion(Region region) {
          Log.i(TAG, "I no longer see an beacon");
        }

        @Override
 public void didDetermineStateForRegion(int state, Region region)
{
          Log.i(TAG, "I have just switched from seeing/not seeing
beacons: "+state);
        }
        });

        try {
           beaconManager.startMonitoringBeaconsInRegion(new
Region("myMonitoringUniqueId", null, null, null));
        } catch (RemoteException e) {   }
    }

}
```

2) Ranging 예제 코드

```
public class RangingActivity extends Activity implements BeaconConsumer {
            protected static final String TAG = "RangingActivity";
            private BeaconManager beaconManager =
        BeaconManager.getInstanceForApplication(this);

            @Override
            protected void onCreate(Bundle savedInstanceState) {
                super.onCreate(savedInstanceState);
                setContentView(R.layout.activity_ranging);
                beaconManager.bind(this);
            }
            @Override
            protected void onDestroy() {
                super.onDestroy();
                beaconManager.unbind(this);
            }
            @Override
            public void onBeaconServiceConnect() {
                beaconManager.setRangeNotifier(new RangeNotifier() {
                @Override
                public void didRangeBeaconsInRegion(Collection<Beacon>
        beacons, Region region) {
                    if (beacons.size() > 0) {
                        Log.i(TAG, "The first beacon I see is about
        "+beacons.iterator().next().getDistance()+" meters away.");
                    }
                }
                });

                try {
                    beaconManager.startRangingBeaconsInRegion(new
        Region("myRangingUniqueId", null, null, null));
                } catch (RemoteException e) {   }
            }
        }
```

3) Application class 예제 코드

- Application을 상속하는 클래스를 생성합니다.

```
import android.app.Application;
import android.content.Intent;
import android.util.Log;
```

```
import org.ccbeacon.beacon.startup.BootstrapNotifier;
import org.ccbeacon.beacon.startup.RegionBootstrap;
import org.ccbeacon.beacon.Region;

public class MyApplication extends Application implements BootstrapNotifier {
    private static final String TAG = ".MyApplication";
    private RegionBootstrap regionBootstrap;

    @Override
    public void onCreate() {
            super.onCreate();
        Log.d(TAG, "App started up");
        // wake up the app when any beacon is seen (you can specify specific id
filers in the parameters below)
            Region region = new Region("com.example.myapp.boostrapRegion", null,
null, null);
            regionBootstrap = new RegionBootstrap(this, region);
    }

    @Override
    public void didDetermineStateForRegion(int arg0, Region arg1) {
        // Don't care
    }

    @Override
    public void didEnterRegion(Region arg0) {
        Log.d(TAG, "Got a didEnterRegion call");
        // This call to disable will make it so the activity below only gets
launched the first time a beacon is seen (until the next time the app is
launched)
        // if you want the Activity to launch every single time beacons come into
view, remove this call.
        regionBootstrap.disable();
        Intent intent = new Intent(this, MonitoringActivity.class);
        // IMPORTANT: in the AndroidManifest.xml definition of this activity, you
must set android:launchMode="singleInstance" or you will get two instances
        // created when a user launches the activity manually and it gets launched
from here.
        intent.setFlags(Intent.FLAG_ACTIVITY_NEW_TASK);
        this.startActivity(intent);
    }

    @Override
    public void didExitRegion(Region arg0) {
```

```
        // Don't care
    }
}
```

4) 배터리 절전 예제 코드

- MyApplication class에 추가합니다.

```
public class MyApplication extends Application implements BootstrapNotifier {
    private BackgroundPowerSaver backgroundPowerSaver;

    public void onCreate() {
        super.onCreate();
        // Simply constructing this class and holding a reference to it in your
custom Application class
        // enables auto battery saving of about 60%
        backgroundPowerSaver = new BackgroundPowerSaver(this);
    }
}
```