

# Alert Notification

Date	17 November 2022
Team ID	PNT2022TMID17163
Project Name	Project - IoT based safety gadget for ChildSafety Monitoring and Notification
TEAM LEADER	AISHWARIYA S
TEAM MEMBERS	ABIRAMI R ABITHA G ANNA POORANI M HARSITA K

```
package com.example.geofence;
```

```
import android.content.BroadcastReceiver;
```

```
import android.content.Context;
```

```
import android.content.Intent;
```

```
import android.location.Location;
```

```
import android.os.CountDownTimer;
```

```
import android.util.Log;
```

```
import android.widget.Toast;
```

```
import com.google.android.gms.location.Geofence;
import com.google.android.gms.location.GeofencingEvent;

import java.util.List;
import android.os.Handler;

public class GeofenceBroadcastReceiver extends BroadcastReceiver {

    private static final String TAG = "GeofenceBroadcastReceiv";

    @Override
    public void onReceive(Context context, Intent intent) {
        // TODO: This method is called when the BroadcastReceiver is receiving
        // an Intent broadcast
        //.
        /*Toast.makeText(context, "GEOFENCE_ENTERED", Toast.LENGTH_SHORT).show();

        final Toast mToastToShow;
        int toastDurationInMilliseconds = 1200000;
        mToastToShow = Toast.makeText(context, "GEOFENCE_EXITED", Toast.LENGTH_LONG);

        // Set the countdown to display the toast
        CountdownTimer toastCountDown;
        toastCountDown = new CountdownTimer(toastDurationInMilliseconds, 100000) {
            public void onTick(long millisUntilFinished) {
                mToastToShow.show();
            }
        }
```

```
public void onFinish() {  
    mToastToShow.cancel();  
}  
};
```

```
// Show the toast and starts the countdown  
mToastToShow.show();  
toastCountDown.start();*/
```

```
NotificationHelper notificationHelper = new NotificationHelper(context);  
notificationHelper.sendHighPriorityNotification("GEOFENCE_TRANSITION_ENTER", "",  
MapsActivity.class);
```

```
GeofencingEvent geofencingEvent = GeofencingEvent.fromIntent(intent);
```

```
if (geofencingEvent.hasError()) {  
    Log.d(TAG, "onReceive: Error receiving geofence event...");  
    return;  
}
```

```
List<Geofence> geofenceList = geofencingEvent.getTriggeringGeofences();  
for (Geofence geofence: geofenceList) {  
    Log.d(TAG, "onReceive: " + geofence.getRequestId());  
}
```

```
// Location location = geofencingEvent.getTriggeringLocation();  
int transitionType = geofencingEvent.getGeofenceTransition();
```

```
switch (transitionType) {  
    case Geofence.GEOFENCE_TRANSITION_ENTER:
```

```
        notificationHelper.sendHighPriorityNotification("Entered the Location", "",  
MapsActivity.class);
```

```
        break;
```

```
    case Geofence.GEOFENCE_TRANSITION_EXIT:
```

```
        notificationHelper.sendHighPriorityNotification("Exited the Location ", "",  
MapsActivity.class);
```

```
        break;
```

```
    }
```

```
}
```

```
}
```