

```
package com.example.covid_19alertapp.services;
```

```
import android.app.Service;
```

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

```
import android.content.SharedPreferences;
```

```
import android.os.IBinder;
```

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

```
import androidx.annotation.Nullable;
```

```
import androidx.work.WorkManager;
```

```
import com.example.covid_19alertapp.activities.TrackerSettingsActivity;
```

```
import com.example.covid_19alertapp.extras.Constants;
```

```
import com.example.covid_19alertapp.extras.LocationFetch;
```

```
import com.example.covid_19alertapp.extras.LogTags;
```

```
import com.example.covid_19alertapp.extras.Notifications;
```

```
import com.example.covid_19alertapp.sharedPreferences.SettingsSharedPreferences;
```

```
public class BackgroundLocationTracker extends Service {
```

```
    /*
```

```
    service to track location on background
```

```
    */
```

```
    @Nullable
```

```
    @Override
```

```
    public IBinder onBind(Intent intent) {
```

```
        return null;
```

```
    }
```

```
    @Override
```

```

public int onStartCommand(Intent intent, int flags, int startId) {

    //TODO: (check on upgraded android api) show notification to notify user of tracking
    Notifications.createNotificationChannel(this);

    //remove existing prompt tracker notification
    Notifications.removeNotification(
        Constants.PromptTrackerNotification_ID,
        this
    );

    // TrackerSettingsActivity as notification tap intent
    Intent notificationIntent = new Intent(this, TrackerSettingsActivity.class);

    //show tracking notification
    try {
        //since api26 need to use startForeground() to run services
        startForeground(
            1,
            Notifications.showNotification(
                Constants.TrackingLocationNotification_ID,
                this,
                notificationIntent,
                false)
        );
    } catch (Exception e){
        // probably older api version or no foreground permission
        Notifications.showNotification(
            Constants.TrackingLocationNotification_ID,

```

```

        this,

        notificationIntent,

        true

    );

    Log.d(LogTags.Service_TAG, "onStartCommand: "+e.getMessage());
}

// set tracking settings preference to true
SettingsSharedPreferences.setLocationTrackerState(this, true);

//TODO: (check on upgraded API!)
LocationFetch.setup(getApplicationContext());
LocationFetch.startLocationUpdates();

Log.d(LogTags.Service_TAG, "onStartCommand: service started!");

return START_STICKY;
}

@Override
public void onDestroy() {

    //remove the tracking notifier

    Notifications.removeNotification(

        Constants.TrackingLocationNotification_ID,

        this

    );
}

```

```
//stop location updates
```

```
LocationFetch.stopLocationUpdates();
```

```
Log.d(LogTags.Service_TAG, "onDestroy: service destroyed!");
```

```
}
```

```
}
```