Intent Services

11 March 2018 8:23

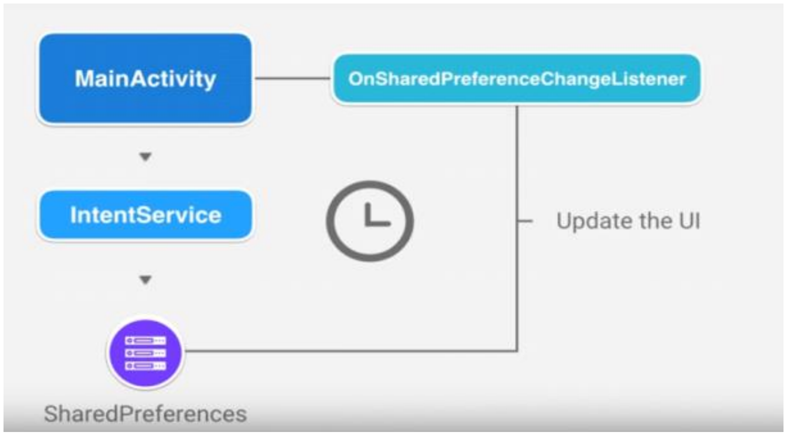
These are implemented to handle background running tasks as a service

In the hydration app where we are incrementing water count by tapping on the image, that increment can be achieved by shared preferences which will keep record of the count **and is much fast.**

But still we are trying to implement intent services here cauz it can be required in a case where counter has to be updated on a remote database keeping all our health records.

That can be a time consuming task which needs to be done in background

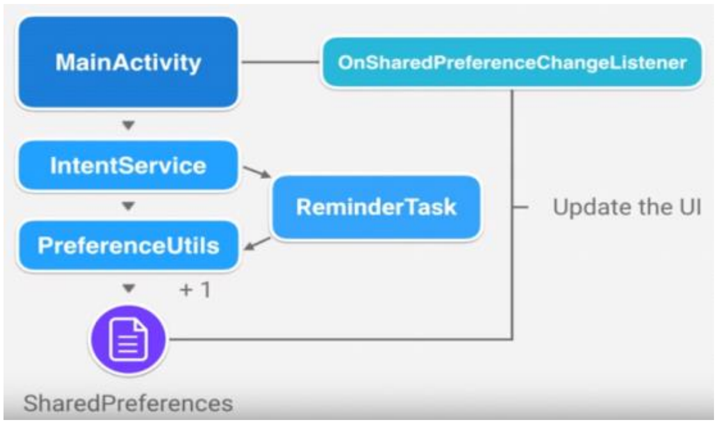
Similarly updating weather data in background also needs to be done in background cauz server requests can be time consuming and might outlast activity lifecycle time.



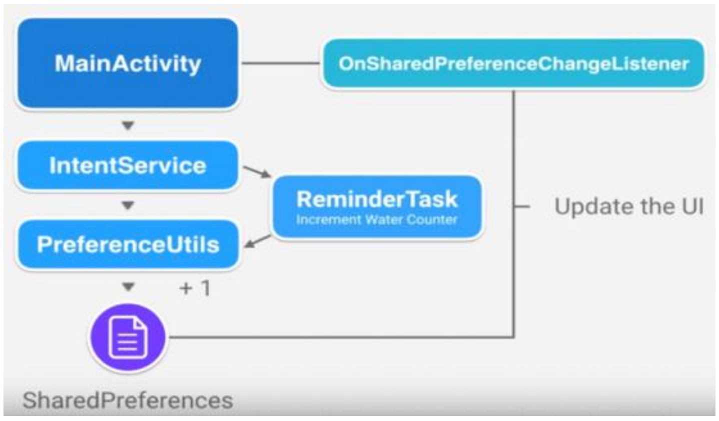
CODE IMPLEMENTATION

- ⭐ **Intent Services handle many tasks in the background. So all tasks should be kept in one place.**
- ⭐ **Here we are defining all tasks under a single roof - Reminder task**

You can start implementing by creating a helper class to keep code organised and modular.

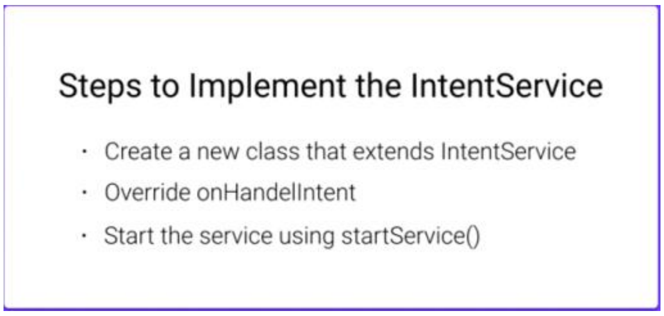


Reminder task will define all tasks running in background for the app
Tasks can be multiple as well as just a single one
In our case we are taking a single task for now - incrementing the water count



TASK EXECUTION ->

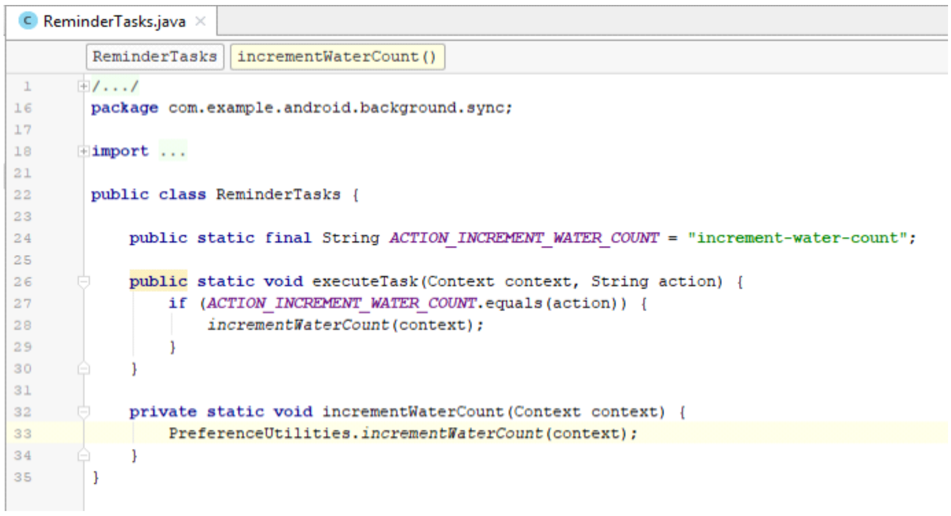
- Activity will give a start to intent service
- Service will execute the increment task
- This task will use PreferenceUtils class to increment counter in shared preferences (SP will keep track of counter all the time)



Steps to Implement the IntentService

- Create a new class that extends IntentService
- Override onHandelIntent
- Start the service using startService()

CODE



ReminderTasks.java

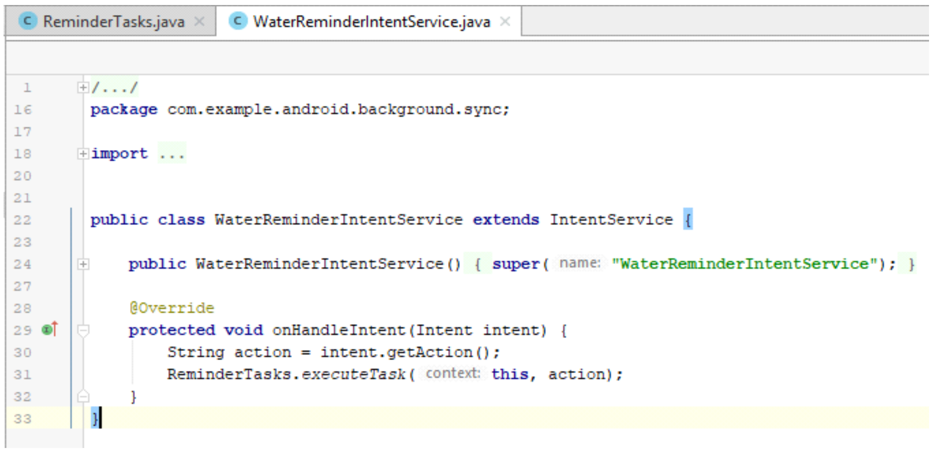
ReminderTasks | incrementWaterCount()

```java
/.../
package com.example.android.background.sync;

import ...

public class ReminderTasks {

    public static final String ACTION_INCREMENT_WATER_COUNT = "increment-water-count";

    public static void executeTask(Context context, String action) {
        if (ACTION_INCREMENT_WATER_COUNT.equals(action)) {
            incrementWaterCount(context);
        }
    }

    private static void incrementWaterCount(Context context) {
        PreferenceUtilities.incrementWaterCount(context);
    }
}
```

--> First, we give name to our first task as an action name - "increment-water-count"
   This task will be referred always by its action name.
   First intent action that reminder class has to handle

--> Create method that performs incrementation of water count - incrementWaterCount
   It uses PreferenceUtils class to do that which incements the count stored in SP

--> Finally, create executeTask method which will
   execute the increment method only if the received action matches the action name we created
   After all we have to know which task we need to perform in case of multiple tasks and this method will keep a check on that



ReminderTasks.java | WaterReminderIntentService.java

```java
/.../
package com.example.android.background.sync;

import ...


public class WaterReminderIntentService extends IntentService {

    public WaterReminderIntentService() { super( name: "WaterReminderIntentService"); }

    @Override
    protected void onHandleIntent(Intent intent) {
        String action = intent.getAction();
        ReminderTasks.executeTask( context: this, action);
    }
}
```

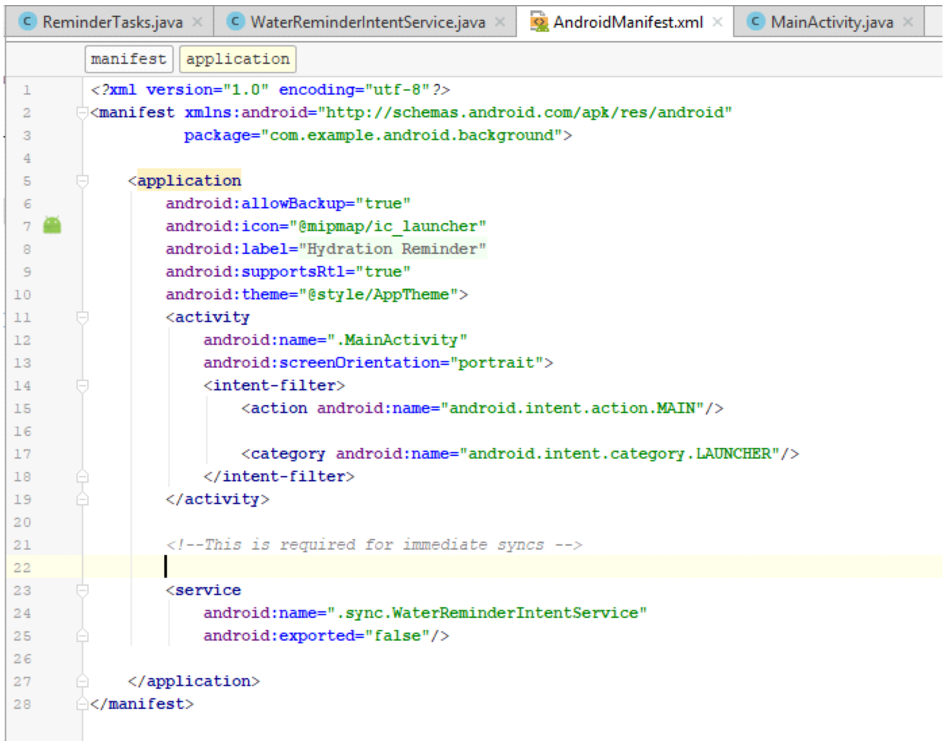
--> Coding the actual intent service

--> Create WaterReminderIntentService class which must extend IntentService

--> Fill in the constructor with name of service - we take name of class as service here

--> Override inbuilt method - OnhandleIntent
   Method that intent service calls on background thread
   We need to know which action we need to execute so get the action from intent first of all - getAction()

Now as we have the task to perform, perform it by calling execute method from our helper class, tell the helper class that this action needs to be performed by sending action name and necessary context



ReminderTasks.java | WaterReminderIntentService.java | AndroidManifest.xml | MainActivity.java

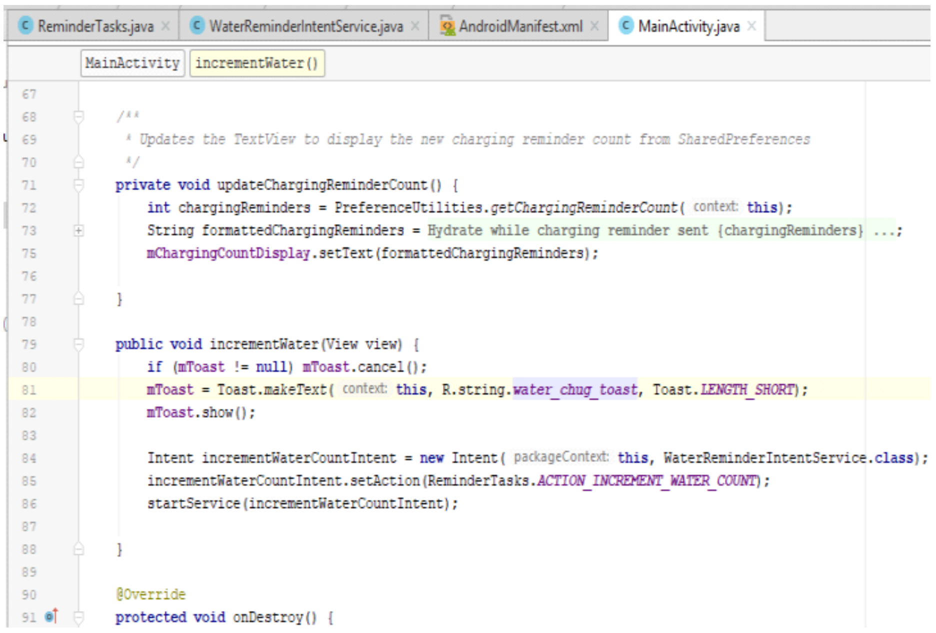
manifest | application

```xml
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
          package="com.example.android.background">

    <application
        android:allowBackup="true"
        android:icon="@mipmap/ic_launcher"
        android:label="Hydration Reminder"
        android:supportsRtl="true"
        android:theme="@style/AppTheme">
        <activity
            android:name=".MainActivity"
            android:screenOrientation="portrait">
            <intent-filter>
                <action android:name="android.intent.action.MAIN"/>

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

        <!--This is required for immediate syncs -->

        <service
            android:name=".sync.WaterReminderIntentService"
            android:exported="false"/>

    </application>
</manifest>
```

--> As any other android component, service needs to be registered in android manifest

--> We'll create a service tag inside the application tag itself

--> Setting exported attribute as false avoids other apps from using our service class, just like in content provider



ReminderTasks.java | WaterReminderIntentService.java | AndroidManifest.xml | MainActivity.java

MainActivity | incrementWater()

```java

    /**
     * Updates the TextView to display the new charging reminder count from SharedPreferences
     */
    private void updateChargingReminderCount() {
        int chargingReminders = PreferenceUtilities.getChargingReminderCount( context: this);
        String formattedChargingReminders = Hydrate while charging reminder sent {chargingReminders} ...;
        mChargingCountDisplay.setText(formattedChargingReminders);

    }

    public void incrementWater(View view) {
        if (mToast != null) mToast.cancel();
        mToast = Toast.makeText( context: this, R.string.water_chug_toast, Toast.LENGTH_SHORT);
        mToast.show();

        Intent incrementWaterCountIntent = new Intent( packageContext: this, WaterReminderIntentService.class);
        incrementWaterCountIntent.setAction(ReminderTasks.ACTION_INCREMENT_WATER_COUNT);
        startService(incrementWaterCountIntent);

    }

    @Override
    protected void onDestroy() {
```

--> Everything done, we just need to start the service now and that will be done in main activity

--> We define an intent for our service - incrementWatrerCountIntent
    Set the action name for it to perform that action - name is taken form our helper class itself where it is defined
    Start the service

--> The method incrementWater(inside which we just started our service), is already set to be called at every click on the cup

--> We have already set the activity to listen to preference changes(change will be increment in count here), so the UI will update on every count to show the changed/incremented count