



Broadcast Receiver

Himadri Parikh

Contents

- Introduction
- Requirements
- Permission
- Designing the app
- Coding in MainActivity.java file

Introduction

- Broadcast Receivers simply respond to broadcast messages from other applications or from the system itself.
- These messages are sometime called events or intents.
- For example, applications can also initiate broadcasts
 - to let other applications know that some data has been downloaded to the device and is available for them to use,
 - so this is broadcast receiver who will intercept this communication and will initiate appropriate action.

Requirements

- Here using broadcast receiver we are trying to make a simple wifiManager app, following are the requirements:
 - Switch
 - Permissions
 - WifiManager
 - BroadcastReceiver
 - Register receiver in Manifest file

Permission

- In Manifest file we need wifi permission as follows:
 - `<uses-permission android:name="android.permission.CHANGE_WIFI_STATE"/>`
 - `<uses-permission android:name="android.permission.ACCESS_WIFI_STATE"/>`
- Designing the app
 - Take a switch and set text as “Wifi is OFF”

Coding in MainActivity.java file

- Define variables: //outside onCreate()
 - `Switch wifiSwitch;`
`WifiManager wifiManager;`
- Binding view and initialisation: //in onCreate()
 - `wifiSwitch = findViewById(R.id.switchWifi);`
 - `wifiManager = (WifiManager)`
`getApplicationContext().getSystemService(Context.WIFI_SERVICE);`

- `setOnCheckedChangeListener: //in onCreate()`
 - `wifiSwitch.setOnCheckedChangeListener(new
CompoundButton.OnCheckedChangeListener() {
 @Override
 public void onCheckedChanged(CompoundButton buttonView, boolean
isChecked) {
 if(isChecked){
 wifiManager.setWifiEnabled(true);
 wifiSwitch.setText("Wifi is ON");
 }
 else {
 wifiManager.setWifiEnabled(false);
 wifiSwitch.setText("Wifi is OFF");
 }
 }
});`

- **Buzzy:** Checking wifiStatusChanged: //in onCreate()
 - `if(wifiManager.isWifiEnabled()){
 wifiSwitch.setChecked(true);
 wifiSwitch.setText("Wifi is ON");
}else {
 wifiManager.setChecked(false);
 wifiSwitch.setText("Wifi is OFF");
}`

BroadcastReceiver

- Create a BroadcastReceiver:

```
• private BroadcastReceiver wifiStateReceiver = new BroadcastReceiver() {  
    @Override  
    public void onReceive(Context context, Intent intent) {  
        int wifiStateExtra = intent.getIntExtra(WifiManager.EXTRA_WIFI_STATE,  
WifiManager.WIFI_STATE_UNKNOWN);  
  
        switch (wifiStateExtra){  
            case WifiManager.WIFI_STATE_ENABLED:  
                wifiSwitch.setChecked(true);  
                wifiSwitch.setText("Wifi is ON");  
                break;  
  
            case WifiManager.WIFI_STATE_DISABLED:  
                wifiSwitch.setChecked(false);  
                wifiSwitch.setText("Wifi is OFF");  
                break;  
        }  
    }  
};
```

- We can remove or disable the code written at Buzzy as the same is included in BroadcastReceiver.
- Register the receiver on onStart():

- **@Override**

- ```
protected void onStart() {
 super.onStart();
```

- ```
        IntentFilter intentFilter = new  
        IntentFilter(WifiManager.WIFI_STATE_CHANGED_ACTION);  
        registerReceiver(wifiStateReceiver, intentFilter);  
    }
```

- Unregister the receiver in onStop():

- **@Override**

- ```
protected void onStop() {
 super.onStop();
 unregisterReceiver(wifiStateReceiver);
}
```

- When the app is on it works in sync with the wifi Manager
- But even if the app is closed it works in sync with the Manager as we have registered our BroadcastReceiver in onStart() method.
- So whenever the app starts and onStart() is called it will check the wifi status and take action on the switch in the app, making it work in sync with the wifi manager.



makeameme.org

Himadri Vrajesh Parikh

[himadrichaudhary1314@gmail.com](mailto:himadrichaudhary1314@gmail.com)

<https://www.linkedin.com/in/himadri-parikh-506219109/>

+91-7990403184