

Flutter: Android Studioda to'laqonli mobil ilova: oddiy messenjer yaratish

Reja:

1. Android Studioda oddiy messenjer yaratish

Messenger **iPC (Inter-Process Communication)** bilan boshqa jarayonda xabarlarni yuborish uchun ishlatiladi va shu bilan mijoz va server o'rtasidagi aloqani ta'minlaydi. Messenger ishlovchiga (handler) ulanadi, shuning uchun barcha ishlar bitta ishlovchining oqimida bo'ladi.

Xizmatni bog'lash uchun Messenger exmeplyar-ga bog'lanishni usulga onBind() o'tkazish va Messenger misolini yaratishda ishlovchining havolasini yuborish kerak. Keling, ishlovchiga ulangan Messengerni ishlatib, boshqa jarayonda ishlaydigan xizmatni yaratishning kichik misolini ko'rib chiqaylik:

```
1 package com.javadevblog.messengerexampleapp;
2 import android.app.Service;
3 import android.content.Intent;
4 import android.os.Bundle;
5 import android.os.Handler;
6 import android.os.IBinder;
7 import android.os.Message;
8 import android.os.Messenger;
9 import android.os.RemoteException;
10 import android.support.annotation.Nullable;
11 import android.widget.Toast;
12 public class SimpleServiceIPC extends Service {
13     public static final int TASK_1 = 1;
14     public static final int TASK_RESPONSE_1 = 2;
15     Messenger messenger = new Messenger(new IncomingHandler());
```

```

16  @Nullable
17  @Override
18  public IBinder onBind(Intent intent) {
19      return messenger.getBinder();
20  }
21  class IncomingHandler extends Handler {
22      @Override
23      public void handleMessage(Message msg) {
24          Message message;
25          Bundle bundle = new Bundle();
26          String messageText;
27          switch (msg.what) {
28              case TASK_1:
29                  messageText = msg.getData().getString("message");
30                  message = Message.obtain(null, TASK_RESPONSE_1);
31                  Toast.ushbusaytga tashrif buyurib , siz qonuniy yoshga
32 To'lganligingizni tasdiqlaysiz va ushbu veb-saytga tashrif buyurganingiz
33 mahalliy qonunchilikning buzilishi emasLENGTH_SHORT).show();
34                  bundle.putString("message_res", messageText);
35                  message.setData(bundle);
36                  Messenger activityMessenger = msg.replyTo;
37                  try {
38                      activityMessenger.send(message);
39                  } catch (RemoteException e) {
40                      e.printStackTrace();
41                  }
42                  break;
43              default:
44                  super.handleMessage(msg);

```

```

45     }
46     }
    }

```

Koddan ko'rinib turibdiki, xizmat Activity bilan muloqot qilish uchun Messengerni ishlatadi. Service va Activity o'rtasidagi barcha ma'lumotlar Bundle ob'ektlarida uzatiladi, bu esa har qanday ma'lumotni uzatish uchun juda qulaydir. Xizmatning o'zi incominghandler sinfidan foydalanadi — qabul qilingan xabarni qayta ishlash uchun Handler merosxo'ri.

Keling, ushbu xizmatni serverga xabar yuborish va Messenger ishlovchisini ishlatib, serverga javob berish uchun faollashtirish yarataylik:

```

1  package com.javadevblog.messengerexampleapp;
2  import android.content.ComponentName;
3  import android.content.Context;
4  import android.content.Intent;
5  import android.content.ServiceConnection;
6  import android.os.Bundle;
7  import android.os.Handler;
8  import android.os.IBinder;
9  import android.os.Message;
10 import android.os.Messenger;
11 import android.os.RemoteException;
12 import android.support.annotation.Nullable;
13 import android.support.v7.app.AppCompatActivity;
14 import android.view.View;
15 import android.widget.Button;
16 import android.widget.EditText;
17 import android.widget.Toast;
18 public class MainActivity extends AppCompatActivity {
19     private final Messenger mActivityMessenger = new Messenger(new

```

```
20  ResponseHandler(this));
21  private Button mButtonSend;
22  private EditText mEditTextMessage;
23  private Messenger mMessenger;
24  private boolean isBound;
25  ServiceConnection serviceConnection = new ServiceConnection() {
26      @Override
27      public void onServiceConnected(ComponentName name, IBinder
28  service) {
29          mMessenger = new Messenger(service);
30          isBound = true;
31      }
32      @Override
33      public void onServiceDisconnected(ComponentName name) {
34          isBound = false;
35          mMessenger = null;
36      }
37  };
38  @Override
39  protected void onCreate(@Nullable Bundle savedInstanceState) {
40      super.onCreate(savedInstanceState);
41      setContentView(R.layout.activity_main);
42      mEditTextMessage = (EditText) findViewById(R.id.et_message);
43      mButtonSend = (Button) findViewById(R.id.btn_send);
44      mButtonSend.setOnClickListener(new View.OnClickListener() {
45          @Override
46          public void onClick(View v) {
47              String messageText = mEditTextMessage.getText().toString();
48              if (messageText.isEmpty()) {
```

```

49         Toast.makeText(MainActivity.bu, " xabar kiriting!",
50 Toast.LENGTH_LONG).show();
51     } else {
52         Message message = Message.obtain(null,
53 SimpleServiceIPC.TASK_1);
54         Bundle bundle = new Bundle();
55         bundle.putString("message", messageText);
56         message.setData(bundle);
57         message.replyTo = mActivityMessenger;
58         try {
59             mMessenger.send(message);
60         } catch (RemoteException e) {
61             e.printStackTrace();
62         }
63     }
64 }
65 });
66 }
67
68 @Override
69 protected void onStart() {
70     super.onStart();
71     if (!isBound) {
72         // start service here
73         Intent intent = new Intent(MainActivity.this, SimpleServiceIPC.class);
74         bindService(intent, serviceConnection,
75 Context.BIND_AUTO_CREATE);
76     }
77 }

```

```

78     @Override
79     protected void onStop() {
80         super.onStop();
81         isBound = false;
82         mMessenger = null;
83     }
84     private static class ResponseHandler extends Handler {
85         private Context mContext;
86         ResponseHandler(Context context) {
87             mContext = context;
88         }
89         @Override
90         public void handleMessage(Message msg) {
91             switch (msg.what) {
92                 case SimpleServiceIPC.TASK_RESPONSE_1:
93                     String result = msg.getData().getString("message_res");
94                     Toast.makeText(mContext, "iPC xizmatidan keldi:" + result,
95 Toast.LENGTH_LONG).show();
96                     break;
97                 default:
98                     super.handleMessage(msg);
99             }
100         }
101     }
    }

```

Ushbu ilovada biz ham bo'lishi kerak ServiceConnection mijoz va server o'rtasidagi aloqani yaratish uchun xizmatga. ResponseHandler-Handler vorisi Messenger'a javobini qayta ishlaydi va foydalanuvchi tajribasini yangilaydi.

Endi ilova bo'limda bizning xizmat qo'shishAndroidManifest.xml:

```
1 <service android:name=".SimpleServiceIPC" android:process=":remote"/>
```

Layout fayliga faqat kirish maydonini va xabarni yuborish uchun tugmani qo'shing:

```
1
```

```
2 <?xml version="1.0" encoding="utf-8"?>
```

```
3 <LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
```

```
4     xmlns:tools="http://schemas.android.com/tools"
```

```
5     android:layout_width="match_parent"
```

```
6     android:layout_height="match_parent"
```

```
7     android:orientation="vertical"
```

```
8     tools:context="com.javadevblog.messengerexampleapp.MainActivity">
```

```
9     <EditText
```

```
10         android:id="@+id/et_message"
```

```
11         android:layout_width="match_parent"
```

```
12         android:layout_height="wrap_content"
```

```
13         android:hint="xabarni kiriting..." />
```

```
14     <Button
```

```
15         android:id="@+id/btn_send"
```

```
16         android:layout_width="match_parent"
```

```
17         android:layout_height="wrap_content"
```

```
18         android:layout_marginTop="10dp"
```

```
19         android:background="@color/colorPrimaryDark"
```

```
20         android:text="yuborish"
```

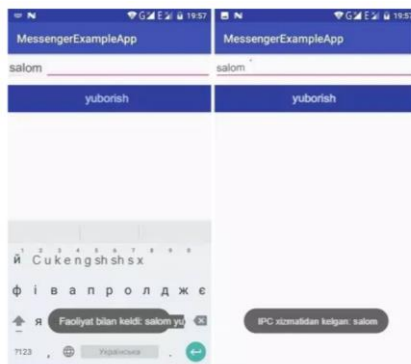
```
21         android:textColor="#fff"/>
```

```
22
```

```
23 </LinearLayout>
```

```
24
```

Endi dasturimizni ishga tushiramiz va natijani ko'rib chiqamiz:



Ko'rib turganingizdek, Messenger server va mijoz o'rtasida muloqot qilishning ajoyib usuli.

Nazorat savollari

1. Messenjer deganda nimani tushunasiz?
2. Android Studioda oddiy messenjer yaratish ketma-ketligini tushuntiring?