

10-ЛЕКЦИЯ. ОБМЕН СООБЩЕНИЯМИ

10.1.Сервисы для управления сообщениями

Обмен сообщениями между облаком и Android устройством (C2DM) – это сервис, который позволяет разработчикам отправлять данные с серверов в приложения на Android устройствах. Сервис предоставляет простой и легкий механизм, который могут использовать сервера для того, чтобы сообщить мобильным приложениям о связи с сервером напрямую для получения обновлений приложения или данных пользователя. C2DM сервис управляет всеми аспектами организации очередей сообщений и доставки к целевому приложению, запущенному на целевом устройстве.

Главные характеристики C2DM:

1. Он позволяет сторонним серверам приложений отправлять небольшие сообщения своим Android приложениям. Сервис обмена сообщениями не предназначен для отправки большого количества пользовательских данных через сообщения. Напротив, он должен использоваться для сообщения приложению, что есть новые данные на сервере, и что приложение может забрать их.
2. Приложение на Android устройстве не нужно запускать для получения сообщений. Система запустит приложение через целевую трансляцию, когда придет сообщение, если приложение установлено с соответствующими приемником трансляции и разрешениями.
3. Он использует существующее соединение для сервисов Google. Это требует от пользователей установки учетной записи Google на их мобильных устройствах.

10.2.Прием и передача сообщений. Обмен сообщениями через пользовательский интерфейс

SMS (Short Message Service) означает «служба коротких сообщений». Эту службу также часто называют обменом текстовыми сообщениями. В Android SDK поддерживаются функции отправки и получения текстовых сообщений. Для отправки смс используется константа **Intent.ACTION_SEND**. Создадим простейший интерфейс для отправки смс:

```
<LinearLayout
```

```
  xmlns:android="http://schemas.android.com/apk/res/android "  
  xmlns:tools="http://schemas.android.com/tools "  
  android:layout_width="match_parent"  
  android:layout_height="match_parent"  
  android:orientation="vertical">
```

```
<EditText android:id="@+id/number"  
  android:layout_width="match_parent"  
  android:layout_height="wrap_content"  
  android:cursorVisible="true "  
  android:hint="Введите номер "  
  android:editable="true "
```

```

        android:singleLine="true" />
<EditText android:id="@+id/message"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:cursorVisible="true"
    android:hint="Введите сообщение"
    android:editable="true"
    android:singleLine="false" />
<Button android:id="@+id/sms"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:text="Отправить"
    android:onClick="smsSend" />
</LinearLayout>

```

И определим обработчик кнопки в коде activity:

```

package com.example.eugene.telephoneapp;

import android.support.v7.app.ActionBarActivity;
import android.os.Bundle;
import android.view.Menu;
import android.view.MenuItem;
import android.content.Intent;
import android.view.View;
import android.widget.EditText;
import android.net.Uri;

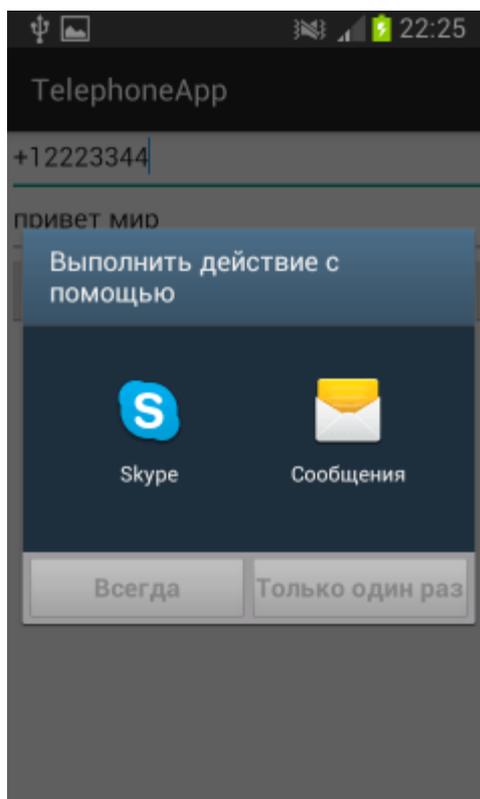
public class MainActivity extends ActionBarActivity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
    }
    public void smsSend(View v) {
        EditText number=(EditText)findViewById(R.id.number);
        EditText message=(EditText)findViewById(R.id.message);
        String toSms="smsto:"+number.getText().toString();
        String messageText= message.getText().toString();
        Intent sms=new Intent(Intent.ACTION_SENDTO, Uri.parse(toSms));

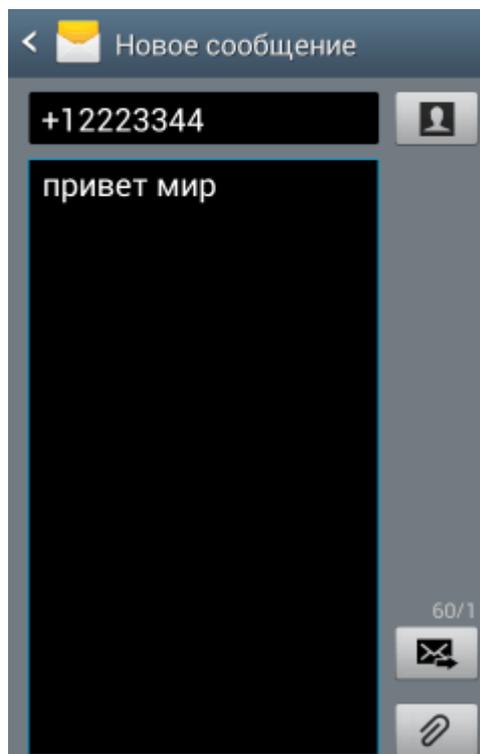
        sms.putExtra("sms_body", messageText);
        startActivity(sms);
    }
}

```

После ввода номера телефона и сообщения нажмем на кнопку отправки, и система нам предоставит выбор между имеющимися программами для отправки сообщений, то есть программа отправки смс и скайп.



После выбора программы можно будет отредактировать и уже потом окончательно отправить сообщение:



Но как и с телефоном мы также можем использовать прямую отправку смс без сторонних программ. Для этого, во-первых, добавим разрешение **SEND_SMS** в файл манифеста:

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

```
package="com.example.eugene.telephoneapp" >  
<uses-permission android:name="android.permission.SEND_SMS" />
```

Исправим код обработчика кнопки:

```
public void dial(View v) {  
  
    EditText number=(EditText)findViewById(R.id.number);  
    EditText message=(EditText)findViewById(R.id.message);  
  
    String numberText = number.getText().toString();  
    String messageText= message.getText().toString();  
  
    SmsManager.getDefault()  
        .sendTextMessage(numberText, null, messageText.toString(), null, null);  
}
```

Для отправки используется класс **SmsManager** из пакета *android.telephony*. Его статический метод `getDefault()` возвращает объект данного класса. Для самой отправки применяется метод `sendTextMessage()`, который принимает пять параметров: номер телефона, адрес сервисного центра (в данном случае `null`), сообщение к отправке и два специальных объекта `PendingIntent`, которые показывают статус отправки и доставки (в данном случае также `null`).

Контрольные вопросы:

1. Каким образом функционируют сервисы для управления сообщениями ?
2. Как осуществляется прием и передача сообщений ?
3. Как осуществляется обмен сообщениями через пользовательский интерфейс?