## [Сервисы, приемники сообщений](https://docs.google.com/presentation/d/1wLCJpd5S9O9fHRtSsKd539TpYLFW7YY--gkhqBm2wi8/edit?usp=sharing)

### Цель работы

Познакомится с сервисами Android Studio и научиться создавать приложения принимающие и отправляющие сообщения.

### Задания для выполнения

1. Создайте кнопку, которая будет выводить всплывающее сообщение на экран при нажатии с помощью объекта Toast**.**

Решение этого задания объединено с решением первого дополнительного задания.

1. Создайте проигрыватель PlayService.java для воспроизведения и остановки одной песни.

**MainActivity.java**

package com.example.musicplayer;  
  
import androidx.appcompat.app.AppCompatActivity;  
  
import android.content.Intent;  
import android.os.Bundle;  
import android.view.View;  
import android.widget.Button;  
  
  
public class MainActivity extends AppCompatActivity {  
  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.*activity\_main*);  
  
 final Button btnStart = findViewById(R.id.*button\_start*);  
 final Button btnStop = findViewById(R.id.*button\_stop*);  
  
 btnStart.setOnClickListener(new View.OnClickListener() {  
 @Override  
 public void onClick(View view) {  
 startService(  
 new Intent(MainActivity.this, PlayService.class));  
 }  
 });  
  
 btnStop.setOnClickListener(new View.OnClickListener() {  
 @Override  
 public void onClick(View view) {  
 stopService(  
 new Intent(MainActivity.this, PlayService.class));  
 }  
 });  
 }  
  
}

**PlayService.java**

package com.example.musicplayer;  
  
import android.app.Service;  
import android.content.ContentResolver;  
import android.content.ContentUris;  
import android.content.Intent;  
import android.database.Cursor;  
import android.media.MediaPlayer;  
import android.net.Uri;  
import android.os.IBinder;  
  
import android.widget.Toast;  
  
import java.util.ArrayList;  
import java.util.Timer;  
import java.util.TimerTask;  
  
public class PlayService extends Service {  
  
 private MediaPlayer player;  
  
 @Override  
 public IBinder onBind(Intent intent) {  
 return null;  
 }  
  
 @Override  
 public void onCreate() {  
 super.onCreate();  
 Toast.*makeText*(this, "Служба создана",  
 Toast.*LENGTH\_SHORT*).show();  
 player = MediaPlayer.*create*(this, R.raw.song);  
 player.setLooping(false);  
 }  
  
  
 @Override  
 public int onStartCommand(Intent intent, int flags, int startId) {  
 Toast.*makeText*(this, "Служба запущена",  
 Toast.*LENGTH\_SHORT*).show();  
 player.start();  
 return super.onStartCommand(intent, flags, startId);  
 }  
  
 @Override  
 public void onDestroy() {  
 super.onDestroy();  
 Toast.*makeText*(this, "Служба остановлена",  
 Toast.*LENGTH\_SHORT*).show();  
 player.stop();  
 }  
}

**activity\_main.xml**

*<?*xml version="1.0" encoding="utf-8"*?>*<androidx.constraintlayout.widget.ConstraintLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 xmlns:app="http://schemas.android.com/apk/res-auto"  
 xmlns:tools="http://schemas.android.com/tools"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 tools:context=".MainActivity">  
  
 <Button  
 android:id="@+id/button\_start"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:text="Play"  
 app:layout\_constraintBottom\_toBottomOf="parent"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintHorizontal\_bias="0.497"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toTopOf="parent"  
 app:layout\_constraintVertical\_bias="0.396" />  
  
 <Button  
 android:id="@+id/button\_stop"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:text="Stop"  
 app:layout\_constraintBottom\_toBottomOf="parent"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintHorizontal\_bias="0.0"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toBottomOf="@+id/button\_start"  
 app:layout\_constraintVertical\_bias="0.0" />  
  
</androidx.constraintlayout.widget.ConstraintLayout>

Изображение выглядит как текст

Автоматически созданное описание

Изображение выглядит как текст

Автоматически созданное описание

Изображение выглядит как текст

Автоматически созданное описание

1. Создайте приложение, которое будет отправлять и принимать сообщения.

**AndroidManifest.xml**

*<?*xml version="1.0" encoding="utf-8"*?>*<manifest xmlns:android="http://schemas.android.com/apk/res/android"  
 package="com.example.broadcasting">  
  
 <application  
 android:allowBackup="true"  
 android:icon="@mipmap/ic\_launcher"  
 android:label="@string/app\_name"  
 android:roundIcon="@mipmap/ic\_launcher\_round"  
 android:supportsRtl="true"  
 android:theme="@style/Theme.Broadcasting">  
  
 <receiver  
 android:name=".MessageReceiver"  
 android:enabled="true"  
 android:exported="true" />  
  
 <activity  
 android:name=".MainActivity"  
 android:exported="true">  
 <intent-filter>  
 <action android:name="android.intent.action.MAIN" />  
  
 <category android:name="android.intent.category.LAUNCHER" />  
 </intent-filter>  
 </activity>  
 </application>  
  
</manifest>

**MainActivity.java**

package com.example.broadcasting;  
  
import androidx.appcompat.app.AppCompatActivity;  
  
import android.content.ComponentName;  
import android.content.Intent;  
import android.os.Bundle;  
import android.view.View;  
  
public class MainActivity extends AppCompatActivity {  
  
 public static final String *MESSAGE* = "Срочно пришлите кота!";  
  
  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.*activity\_main*);  
 }  
  
 public void sendMessage(View view) {  
 Intent intent = new Intent();  
 ComponentName componentName = new ComponentName(this, MessageReceiver.class);  
 intent.putExtra("com.example.broadcasting.broadcast.Message", *MESSAGE*);  
 intent.addFlags(Intent.*FLAG\_INCLUDE\_STOPPED\_PACKAGES*);  
 intent.setComponent(componentName);  
 sendBroadcast(intent);  
 }  
}

**MessageReceiver.java**

package com.example.broadcasting;  
  
import android.content.BroadcastReceiver;  
import android.content.Context;  
import android.content.Intent;  
import android.widget.Toast;  
  
public class MessageReceiver extends BroadcastReceiver {  
  
 @Override  
 public void onReceive(Context context, Intent intent) {  
 Toast.*makeText*(context, "Обнаружено сообщение: " +  
 intent.getStringExtra("com.example.broadcasting.broadcast.Message"),  
 Toast.*LENGTH\_LONG*).show();  
  
 }  
}

**activity\_main.xml**

*<?*xml version="1.0" encoding="utf-8"*?>*<androidx.constraintlayout.widget.ConstraintLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 xmlns:app="http://schemas.android.com/apk/res-auto"  
 xmlns:tools="http://schemas.android.com/tools"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 tools:context=".MainActivity">  
  
 <Button  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:text="Отправить сообщение"  
 app:layout\_constraintBottom\_toBottomOf="parent"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toTopOf="parent"  
 android:onClick="sendMessage" />  
</androidx.constraintlayout.widget.ConstraintLayout>

Изображение выглядит как текст

Автоматически созданное описание

### Контрольные вопросы

1. Чем сервисы отличаются от Activity?

Сервисы не имеют пользовательского интерфейса и нужны в тех случаях, когда не требуется вмешательства пользователя. Сервисы работают в фоновом режиме. Они предназначены для длительного существования, в отличие от активностей, и могут работать, постоянно перезапускаясь, выполняя постоянные задачи или выполняя задачи, требующие много времени. Android даёт службам более высокий приоритет, чем бездействующим активностям.

1. Возможно ли в Toast Notification разместить LinearLayout?

В Toast возможно разместить любую View, в том числе содержащую LinearLayout:

LayoutInflater inflater = getLayoutInflater();

View view = inflater.inflate(R.layout.custom\_toast\_layout,  
(ViewGroup) findViewById(R.id.parent));

Toast toast = new Toast(this);

toast.setView(view);

toast.show();

R.layout.custom\_toast\_layout — layout который мы хотим использовать, а R.id.parent — id родительского элемента в layout.

1. Сколько времени отображается всплывшее сообщение, созданное с помощью объекта Toast?

Всплывающее сообщение, созданное с помощью объекта Toast, выведется на 3 с половиной секунды или на 2 секунды в зависимости от того, какую константу мы передадим в метод makeText(): Toast.LENGTH\_LONG или Toast.LENGTH\_SHORT соответственно.

### Дополнительные задания

1. Создайте кнопку, которая будет выводить всплывающую панель с картинкой и текстом на экран при нажатии с помощью объекта Toast**.**

**MainActivity.java**

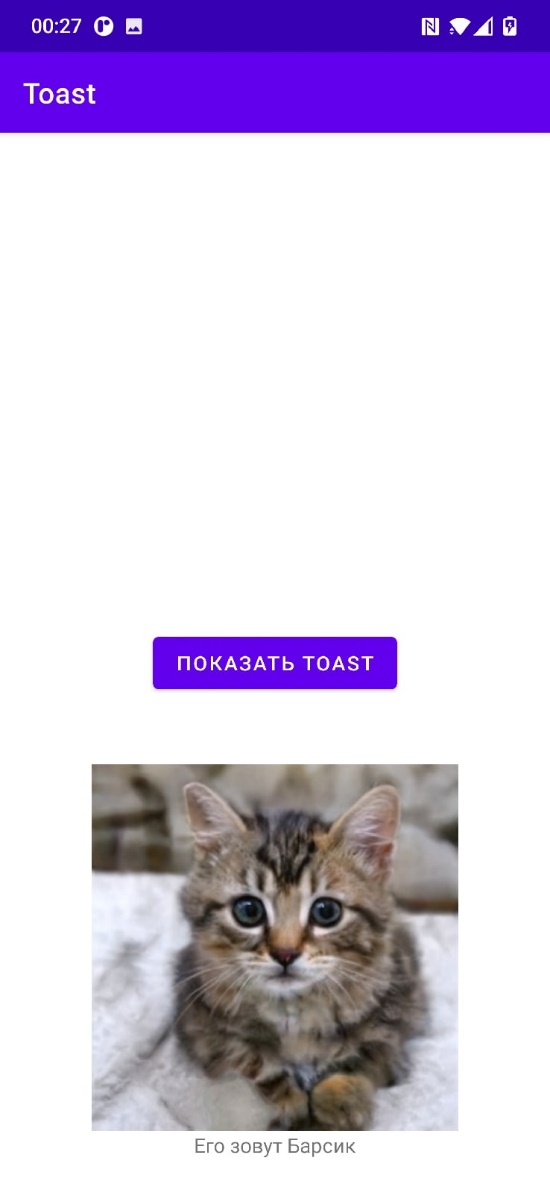
package com.example.toast;  
  
import androidx.appcompat.app.AppCompatActivity;  
  
import android.os.Bundle;  
import android.view.LayoutInflater;  
import android.view.View;  
import android.view.ViewGroup;  
import android.widget.Toast;  
  
public class MainActivity extends AppCompatActivity {  
  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.*activity\_main*);  
 }  
  
 public void onClick(View view) {  
 LayoutInflater inflater = getLayoutInflater();  
 View toastView = inflater.inflate(R.layout.*toast*,  
 (ViewGroup) findViewById(R.id.*toast\_linear\_layout*));  
 Toast toast = new Toast(getApplicationContext());  
 toast.setView(toastView);  
 toast.show();  
 }  
}

**activity\_main.xml**

*<?*xml version="1.0" encoding="utf-8"*?>*<androidx.constraintlayout.widget.ConstraintLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 xmlns:app="http://schemas.android.com/apk/res-auto"  
 xmlns:tools="http://schemas.android.com/tools"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 tools:context=".MainActivity">  
  
 <Button  
 android:id="@+id/button"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:text="Показать Toast"  
 app:layout\_constraintBottom\_toBottomOf="parent"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toTopOf="parent"  
 android:onClick="onClick"/>  
  
</androidx.constraintlayout.widget.ConstraintLayout>

**toast.xml**

*<?*xml version="1.0" encoding="utf-8"*?>*<LinearLayout  
 xmlns:android="http://schemas.android.com/apk/res/android"  
 android:layout\_height="wrap\_content"  
 android:layout\_width="wrap\_content"  
 android:orientation="vertical"  
 android:id="@+id/toast\_linear\_layout">  
  
 <ImageView  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:src="@drawable/cat" />  
  
 <TextView  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:text="Его зовут Барсик"  
 android:gravity="center" />  
  
</LinearLayout>



1. Создайте проигрыватель PlayService.java для воспроизведения списка песен из плей-листа.

Под плей-листом понимаю список всех аудиофайлов, которые лежат во внешней памяти устройства в соответствующих директориях. Код с получением списка плей-листа взят с сайта

<https://developer.android.com/training/data-storage/shared/media>

**MainActivity.java**

package com.example.musicplayer;  
  
import androidx.appcompat.app.AppCompatActivity;  
  
import android.content.Intent;  
import android.os.Bundle;  
import android.view.View;  
import android.widget.Button;  
  
  
public class MainActivity extends AppCompatActivity {  
  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.*activity\_main*);  
  
 final Button btnStart = findViewById(R.id.*button\_start*);  
 final Button btnStop = findViewById(R.id.*button\_stop*);  
  
 btnStart.setOnClickListener(new View.OnClickListener() {  
 @Override  
 public void onClick(View view) {  
 startService(  
 new Intent(MainActivity.this, PlayService.class));  
 }  
 });  
  
 btnStop.setOnClickListener(new View.OnClickListener() {  
 @Override  
 public void onClick(View view) {  
 stopService(  
 new Intent(MainActivity.this, PlayService.class));  
 }  
 });  
 }  
  
}

**PlayService.java**

package com.example.musicplayer;  
  
import android.app.Service;  
import android.content.ContentResolver;  
import android.content.ContentUris;  
import android.content.Intent;  
import android.database.Cursor;  
import android.media.MediaPlayer;  
import android.net.Uri;  
import android.os.IBinder;  
  
import android.widget.Toast;  
  
import java.util.ArrayList;  
import java.util.Timer;  
import java.util.TimerTask;  
  
public class PlayService extends Service {  
  
 private MediaPlayer player;  
 private Timer timer;  
 private final ArrayList<Uri> playlist = new ArrayList<>();  
 int audioIndex = 0;  
  
 @Override  
 public IBinder onBind(Intent intent) {  
 return null;  
 }  
  
 @Override  
 public void onCreate() {  
 super.onCreate();  
  
 ContentResolver contentResolver = getContentResolver();  
  
 Uri uri = android.provider.MediaStore.Audio.Media.*EXTERNAL\_CONTENT\_URI*;  
 Cursor cursor = contentResolver.query(uri, null, null, null, null);  
  
 if (cursor != null && cursor.moveToFirst()) {  
 int idColumn = cursor.getColumnIndex(android.provider.MediaStore.Audio.Media.*\_ID*);  
 do {  
 long thisId = cursor.getLong(idColumn);  
 Uri contentUri = ContentUris.*withAppendedId*(  
 android.provider.MediaStore.Audio.Media.*EXTERNAL\_CONTENT\_URI*, thisId);  
 playlist.add(contentUri);  
 } while (cursor.moveToNext());  
 }  
  
 Toast.*makeText*(this, "Служба создана",  
 Toast.*LENGTH\_SHORT*).show();  
 player = MediaPlayer.*create*(this, playlist.get(audioIndex));  
 player.setLooping(false);  
 }  
  
  
 @Override  
 public int onStartCommand(Intent intent, int flags, int startId) {  
 Toast.*makeText*(this, "Служба запущена",  
 Toast.*LENGTH\_SHORT*).show();  
 player.start();  
 timer = new Timer();  
 if (playlist.size() > 1) {  
 playNext();  
 }  
 return super.onStartCommand(intent, flags, startId);  
 }  
  
 @Override  
 public void onDestroy() {  
 super.onDestroy();  
 Toast.*makeText*(this, "Служба остановлена",  
 Toast.*LENGTH\_SHORT*).show();  
 player.stop();  
 }  
  
 public void playNext() {  
 timer.schedule(new TimerTask() {  
 @Override  
 public void run() {  
 player.reset();  
 player = MediaPlayer.*create*(PlayService.this, playlist.get(++audioIndex));  
 player.start();  
 if (playlist.size() > audioIndex + 1) {  
 playNext();  
 } else {  
 audioIndex = 0;  
 }  
 }  
 }, player.getDuration());  
 }  
}

**activity\_main.xml**

*<?*xml version="1.0" encoding="utf-8"*?>*<androidx.constraintlayout.widget.ConstraintLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 xmlns:app="http://schemas.android.com/apk/res-auto"  
 xmlns:tools="http://schemas.android.com/tools"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 tools:context=".MainActivity">  
  
 <Button  
 android:id="@+id/button\_start"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:text="Play"  
 app:layout\_constraintBottom\_toBottomOf="parent"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintHorizontal\_bias="0.497"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toTopOf="parent"  
 app:layout\_constraintVertical\_bias="0.396" />  
  
 <Button  
 android:id="@+id/button\_stop"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:text="Stop"  
 app:layout\_constraintBottom\_toBottomOf="parent"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintHorizontal\_bias="0.0"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toBottomOf="@+id/button\_start"  
 app:layout\_constraintVertical\_bias="0.0" />  
  
</androidx.constraintlayout.widget.ConstraintLayout>

Изображение выглядит как текст

Автоматически созданное описание

Изображение выглядит как текст

Автоматически созданное описание

Изображение выглядит как текст

Автоматически созданное описание