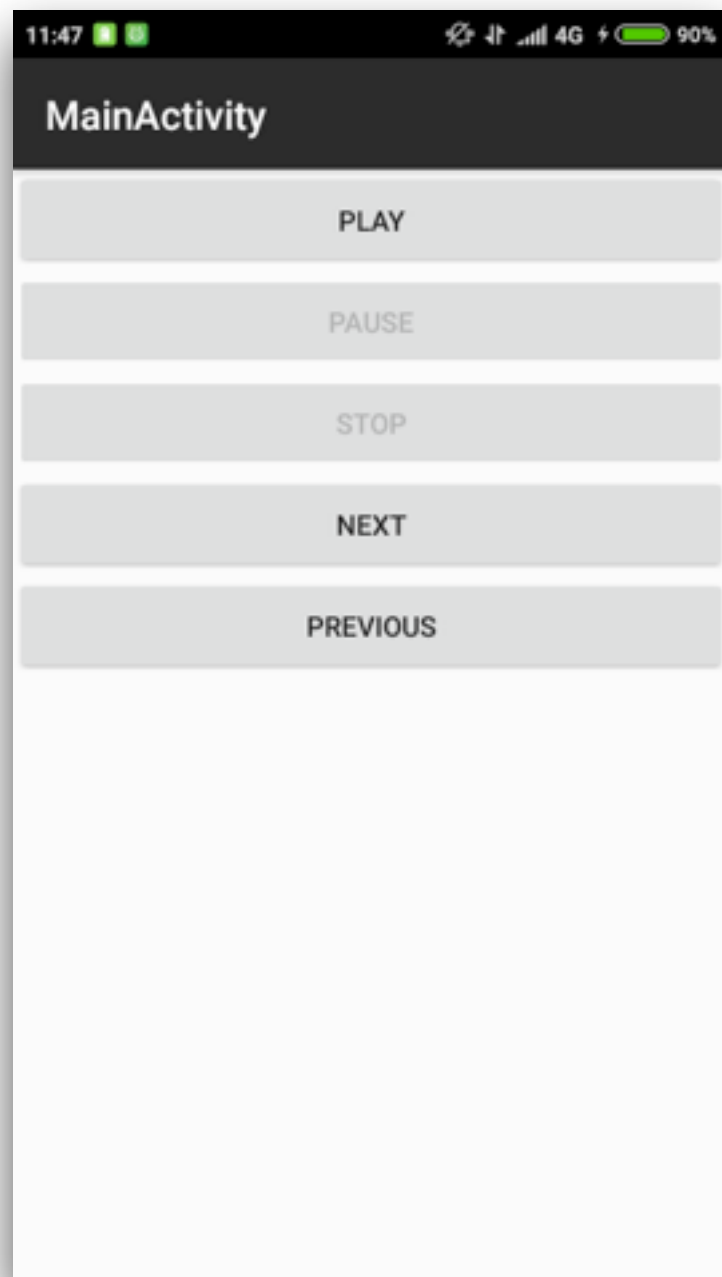
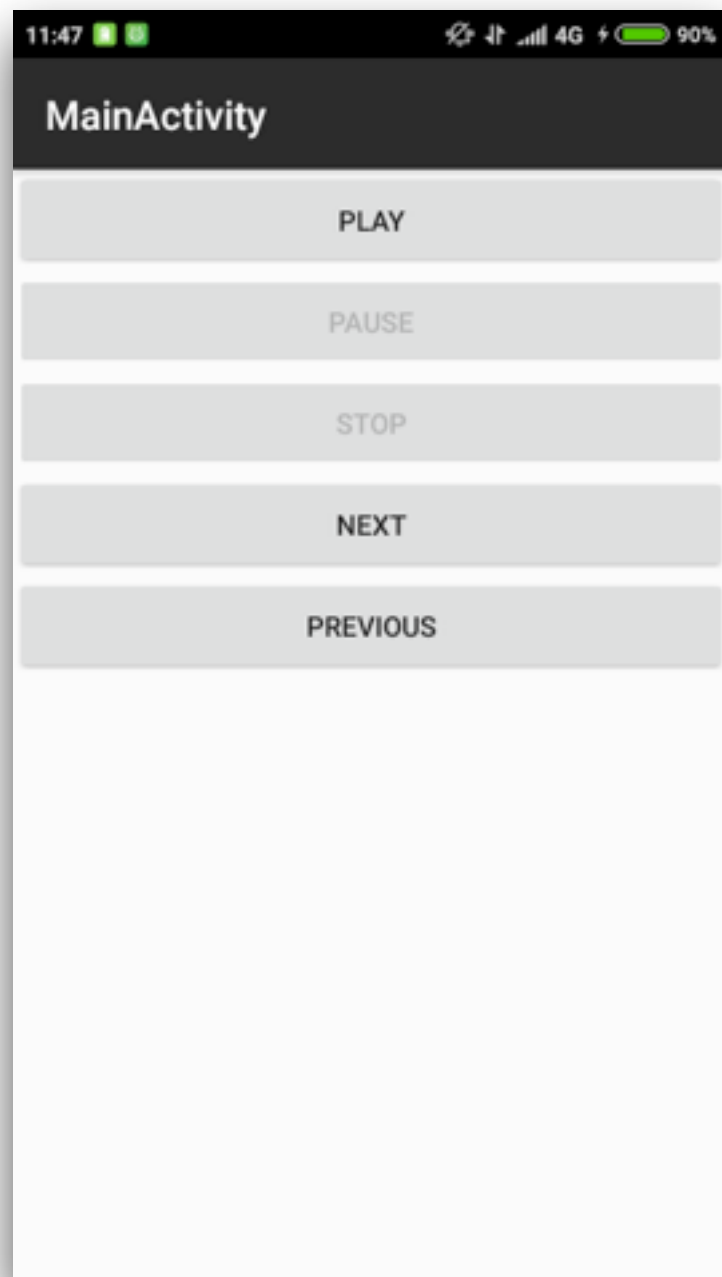




Звук на Андроиде

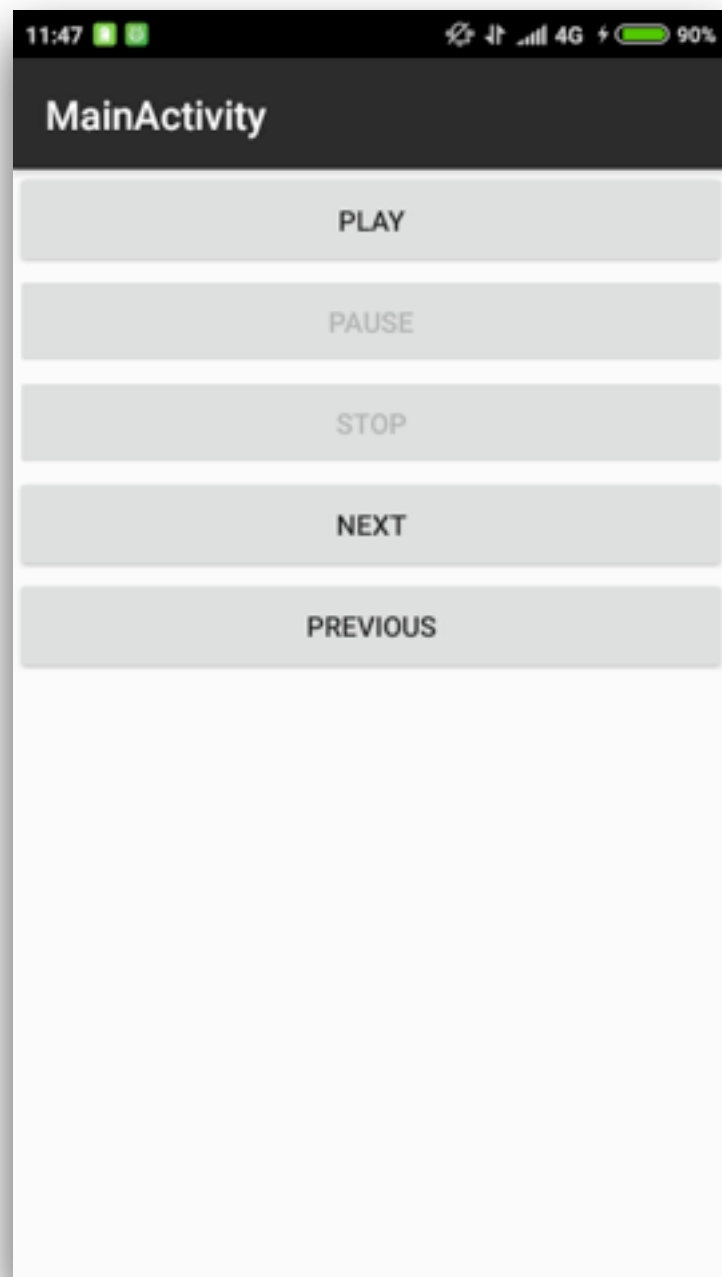




Activity

Обработка событий onClick...

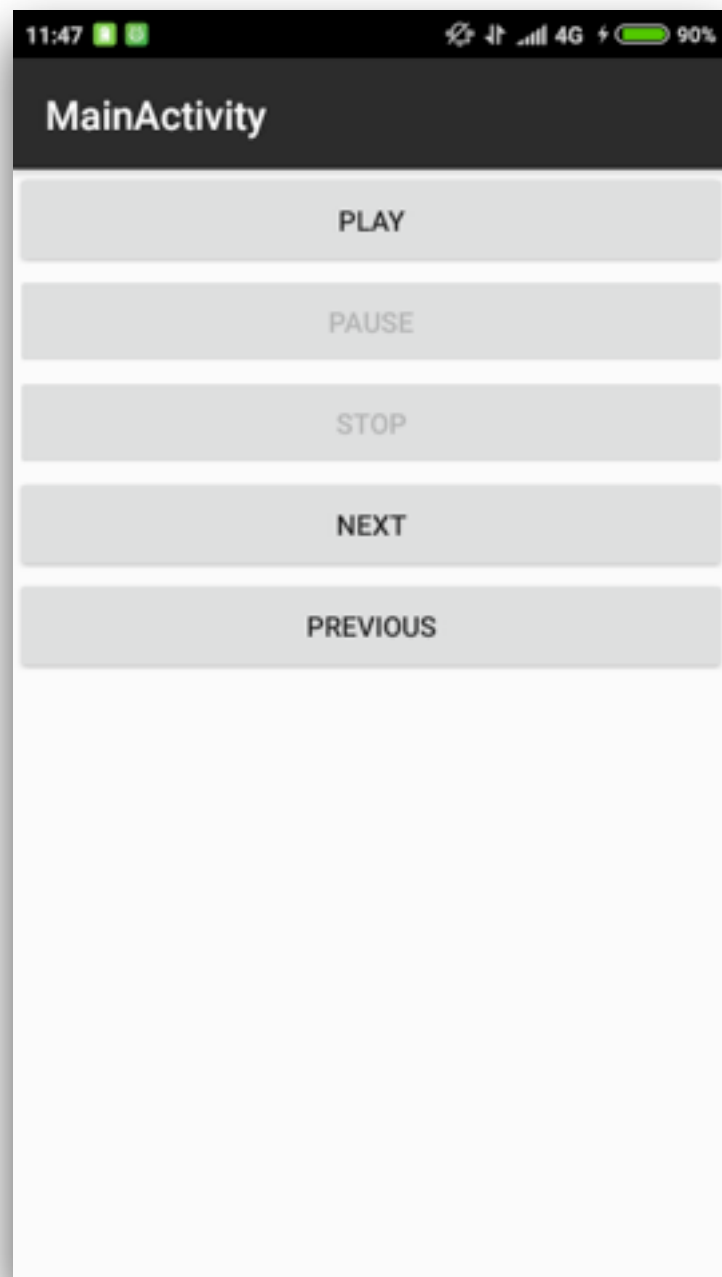




Activity

Обработка событий `onClick...`



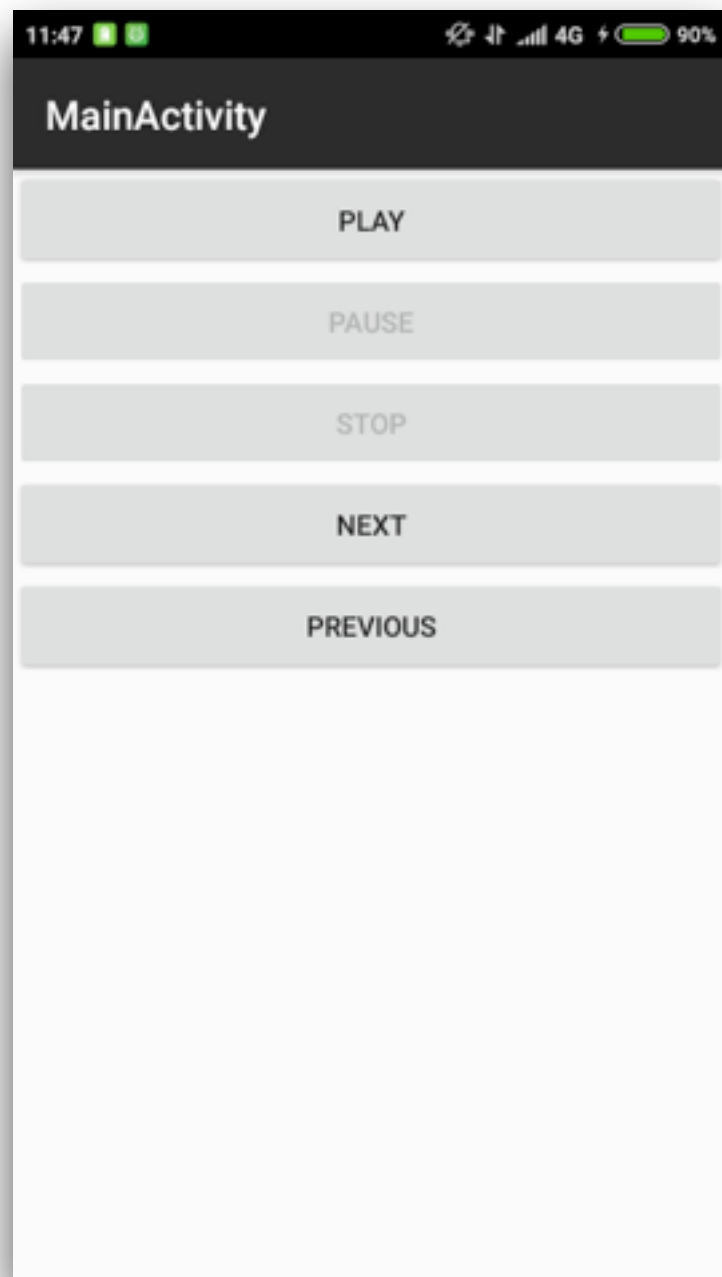


~~Activity~~

~~Обработка событий onClick...~~

Retained fragment



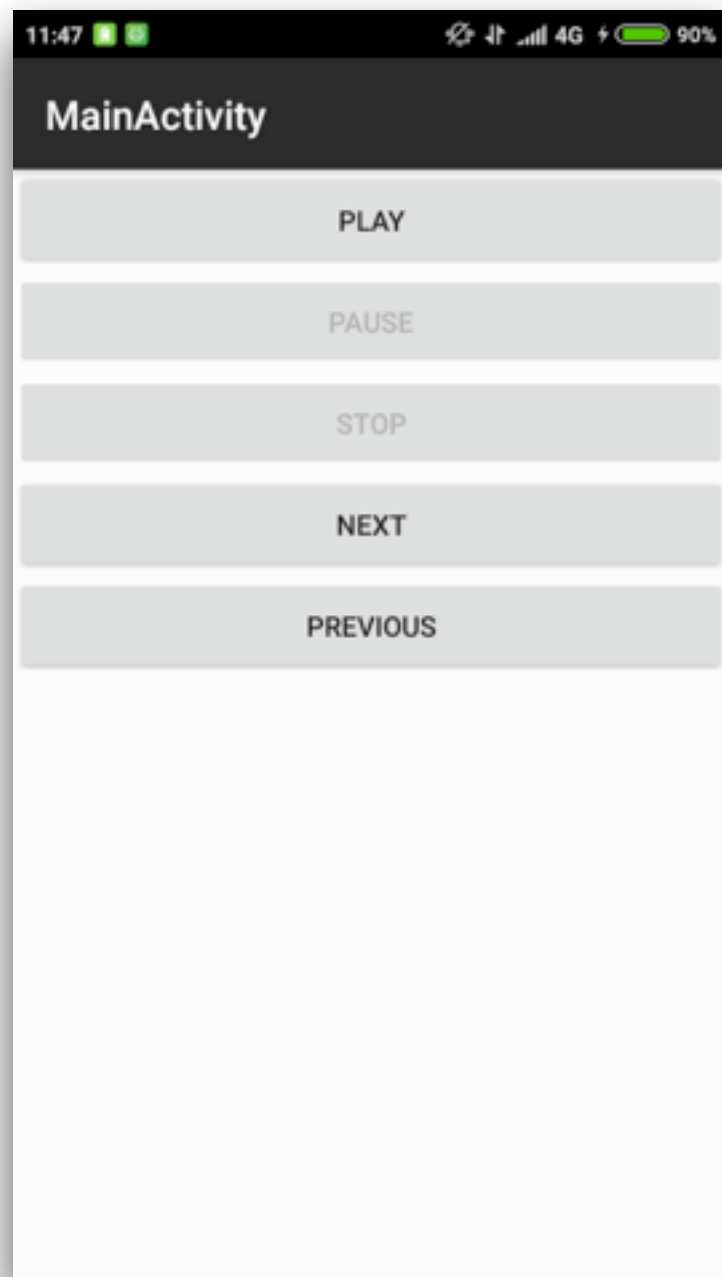


~~Activity~~

~~Обработка событий onClick...~~

~~Retained fragment~~





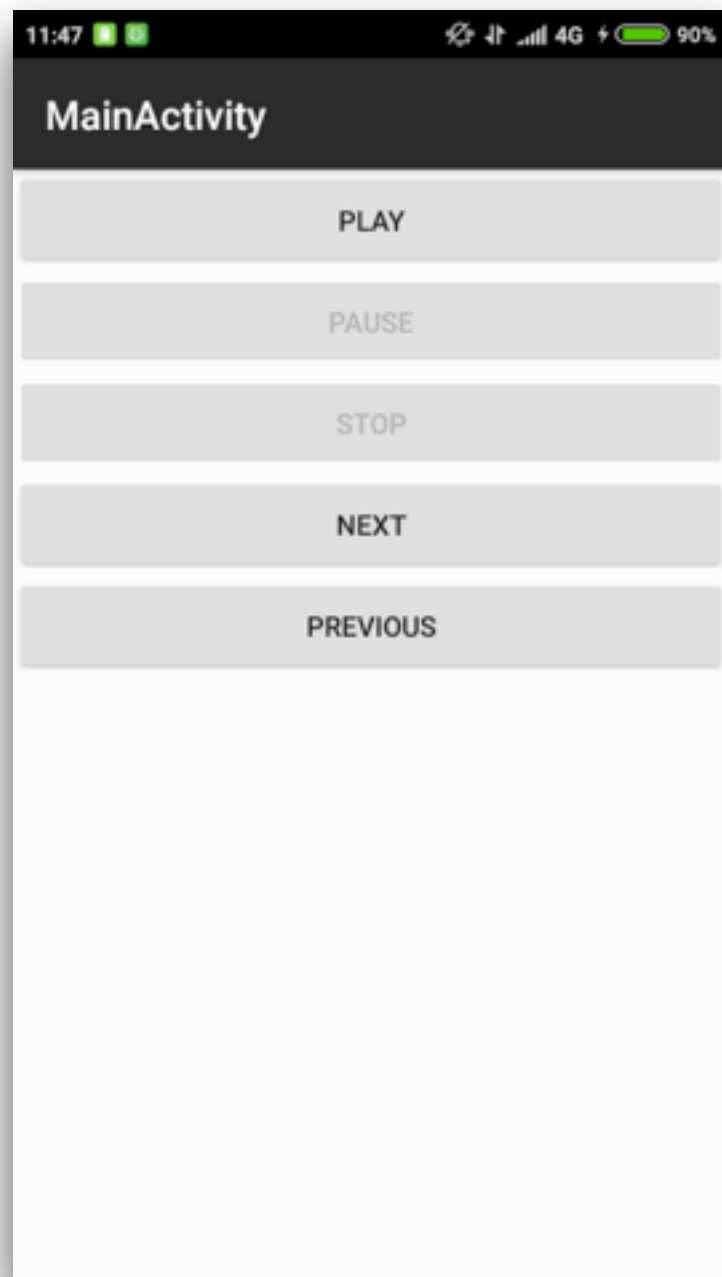
~~Activity~~

~~Обработка событий onClick...~~

~~Retained fragment~~

Service





~~Activity~~

~~Обработка событий onClick...~~

~~Retained fragment~~

Service



MediaSession

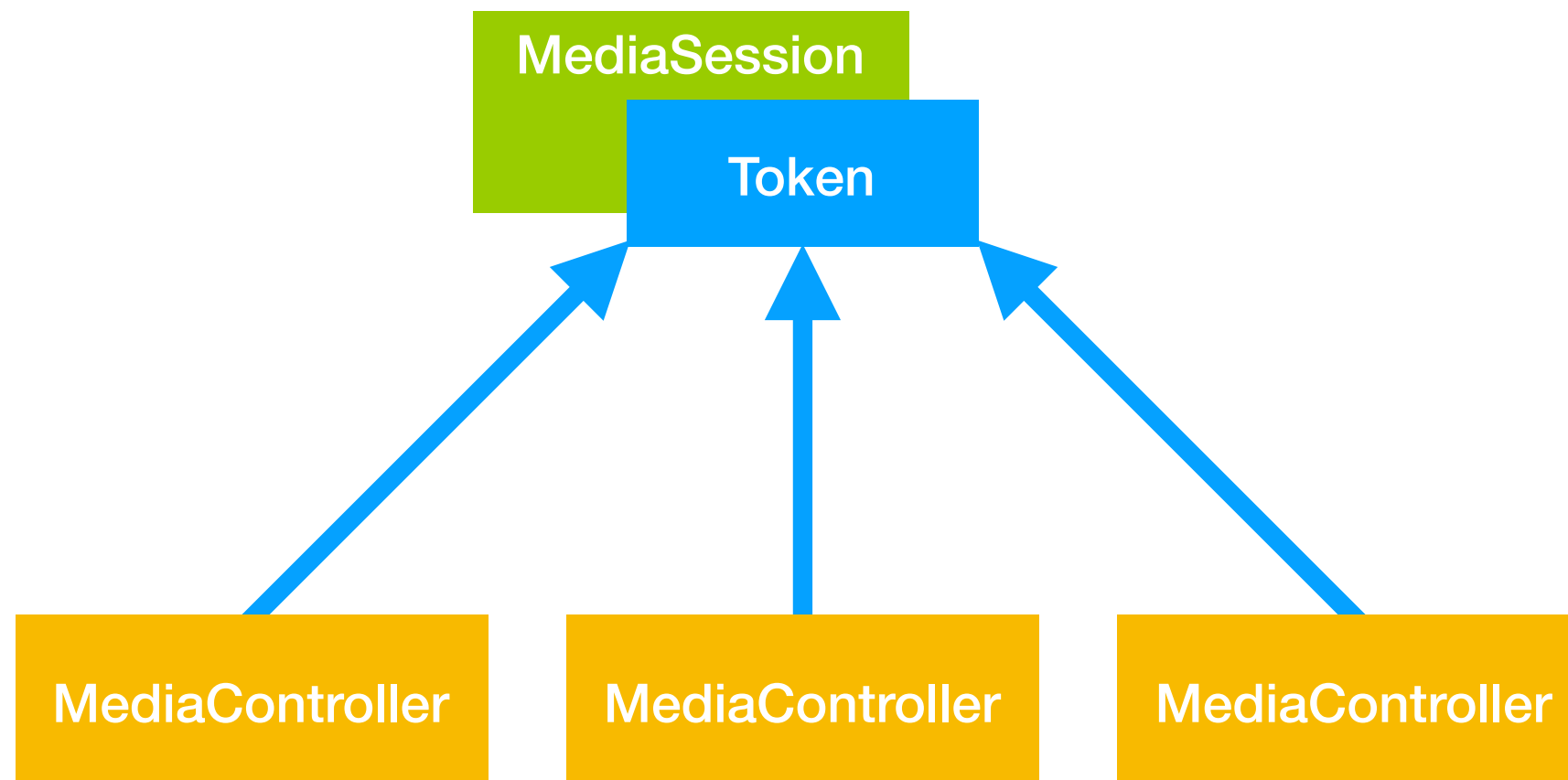
Публикация информации со стороны плеера

<https://developer.android.com/reference/android/media/session/MediaSession.html>

MediaController

Управление плеером со стороны “потребителя”

<https://developer.android.com/reference/android/widget/MediaController.html>



MediaSession.Callback

[onPlay\(\)](#)

[onPause\(\)](#)

[onStop\(\)](#)

[onSkipToNext\(\)](#)

[onSkipToPrevious\(\)](#)

В обработчиках

mediaSession.setPlaybackState(newState);

mediaSession.setMetadata(newMetadata);

mediaSession.setActive(**true/false**);

+ управление самим медиаплеером



Service

```
@Override
public void onCreate() {
    super.onCreate();

    // "PlayerService" – просто tag для отладки
    mediaSession = new MediaSessionCompat(this, "PlayerService");

    // FLAG_HANDLES_MEDIA_BUTTONS – хотим получать события
    // от аппаратных кнопок (например, гарнитуры)
    // FLAG_HANDLES_TRANSPORT_CONTROLS – хотим получать события
    // от кнопок на окне блокировки
    mediaSession.setFlags(MediaSessionCompat.FLAG_HANDLES_MEDIA_BUTTONS
        | MediaSessionCompat.FLAG_HANDLES_TRANSPORT_CONTROLS);

    // Отдаем наши коллбэки
    mediaSession.setCallback(mediaSessionCallback);
}

@Override
public void onDestroy() {
    super.onDestroy();
    mediaSession.release();
}
```



PlaybackState

*// Здесь мы указываем действия, которые собираемся
// обрабатывать в callbacks.
// Например, если мы не укажем ACTION_PAUSE,
// то нажатие на паузу не вызовет onPause.
// ACTION_PLAY_PAUSE обязателен, иначе не будет работать
// управление с Android Wear!*

```
private final PlaybackStateCompat.Builder stateBuilder  
    = new PlaybackStateCompat.Builder().setActions(  
        PlaybackStateCompat.ACTION_PLAY  
        | PlaybackStateCompat.ACTION_STOP  
        | PlaybackStateCompat.ACTION_PAUSE  
        | PlaybackStateCompat.ACTION_PLAY_PAUSE  
        | PlaybackStateCompat.ACTION_SKIP_TO_NEXT  
        | PlaybackStateCompat.ACTION_SKIP_TO_PREVIOUS);
```



PlaybackState

```
mediaSession.setPlaybackState(  
    stateBuilder.setState(  
        PlaybackStateCompat.STATE_PLAYING,  
        PlaybackStateCompat.PLAYBACK_POSITION_UNKNOWN,  
        1 /* playbackSpeed */)   
        .build());
```

Примеры состояний

```
PlaybackStateCompat.STATE_PLAYING  
PlaybackStateCompat.STATE_PAUSED  
PlaybackStateCompat.STATE_SKIPPING_TO_NEXT  
PlaybackStateCompat.STATE_SKIPPING_TO_PREVIOUS
```



Metadata

metadataBuilder

```
.putBitmap(MediaMetadataCompat.METADATA_KEY_ART, bitmap)  
.putString(MediaMetadataCompat.METADATA_KEY_TITLE, title)  
.putString(MediaMetadataCompat.METADATA_KEY_ALBUM, artist)  
.putLong(MediaMetadataCompat.METADATA_KEY_DURATION, duration)  
.build();
```



setActive

Сообщаем системе, что наше приложение - “главный” плеер



onPlay	onPause	onStop	onSkip...
setActive(true)		setActive(false)	
setMetadata			setMetadata
setPlaybackState	setPlaybackState	setPlaybackState	



mediaSession.getSessionToken()

@Nullable

@Override

```
public IBinder onBind(Intent intent) {  
    return new PlayerServiceBinder();  
}
```

```
public class PlayerServiceBinder extends Binder {  
    public MediaSessionCompat.Token getMediaSessionToken() {  
        return mediaSession.getSessionToken();  
    }  
}
```



MediaController

Создается вручную или через [MediaSessionManager](#)

[getMetadata\(\)](#)

[getPlaybackState\(\)](#)

[MediaController.Callback](#)

[onMetadataChanged\(MediaMetadata metadata\)](#)

[onPlaybackStateChanged\(PlaybackState state\)](#)

[MediaController.TransportControls](#)

[play\(\)](#)

[pause\(\)](#)

[stop\(\)](#)

[skipToNext\(\)](#)

[skipToPrevious\(\)](#)



MediaController

```
mediaController = new MediaControllerCompat(  
    context, playerServiceBinder.getMediaSessionToken());  
  
mediaController.registerCallback(  
    new MediaControllerCompat.Callback() {  
        @Override  
        public void onPlaybackStateChanged(PlaybackStateCompat state) {  
            boolean playing  
                = state.getState() == PlaybackStateCompat.STATE_PLAYING;  
            playButton.setEnabled(!playing);  
            pauseButton.setEnabled(playing);  
            stopButton.setEnabled(playing);  
        }  
    });
```

```
mediaController.getTransportControls().play();  
mediaController.getTransportControls().pause();  
mediaController.getTransportControls().stop();
```



MediaButtonReceiver

```
<receiver  
  android:name="android.support.v4.media.session.MediaButtonReceiver">  
  <intent-filter>  
    <action android:name="android.intent.action.MEDIA_BUTTON" />  
  </intent-filter>  
</receiver>
```

```
<service  
  android:name=".service.PlayerService"  
  android:exported="false">  
  <intent-filter>  
    <action android:name="android.intent.action.MEDIA_BUTTON" />  
  </intent-filter>  
</service>
```

@Override

```
public int onStartCommand(Intent intent, int flags, int startId) {  
    MediaButtonReceiver.handleIntent(mediaSession, intent);  
    return super.onStartCommand(intent, flags, startId);  
}
```



В случае API >= 21

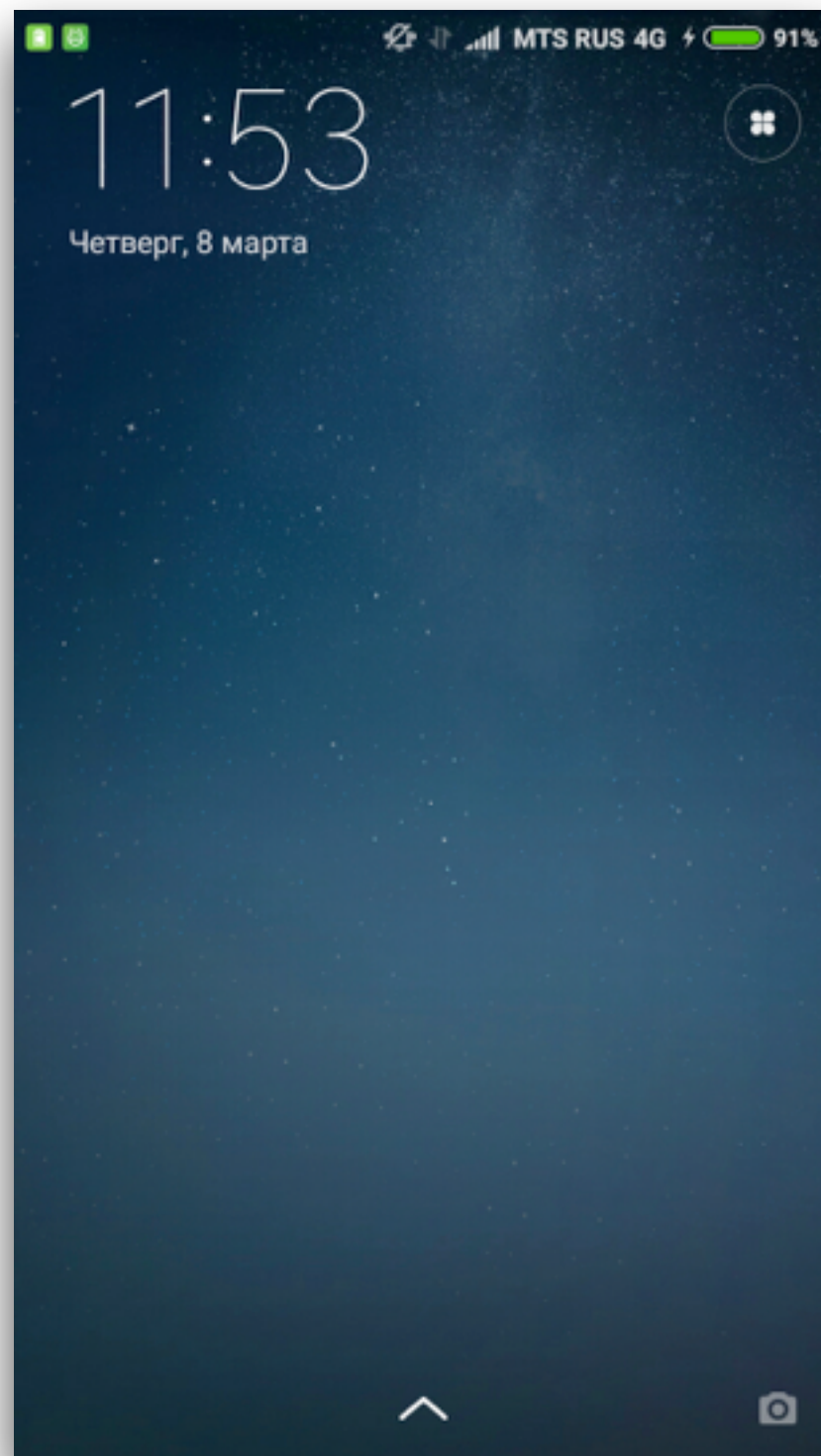
система автоматически “цепляется” к активному MediaSession, но неактивные сессии все-равно пробуждаются бродкастом

```
Intent mediaButtonIntent = new Intent(Intent.ACTION_MEDIA_BUTTON,  
    null, appContext, MediaButtonReceiver.class);  
mediaSession.setMediaButtonReceiver(  
    PendingIntent.getBroadcast(appContext,  
        0, mediaButtonIntent, 0));
```

На системах с API < 21 метод setMediaButtonReceiver ничего не делает

Ну наконец-то, смотрим что получилось...





AudioFocus

Получаем через системный сервис [AudioManager](#)

```
(AudioManager) getSystemService(Context.AUDIO_SERVICE);
```

```
requestAudioFocus(listener, ...)
```

AUDIOFOCUS_REQUEST_GRANTED

AUDIOFOCUS_REQUEST_FAILED

```
abandonAudioFocus(listener)
```

[AudioManager.OnAudioFocusChangeListener](#)

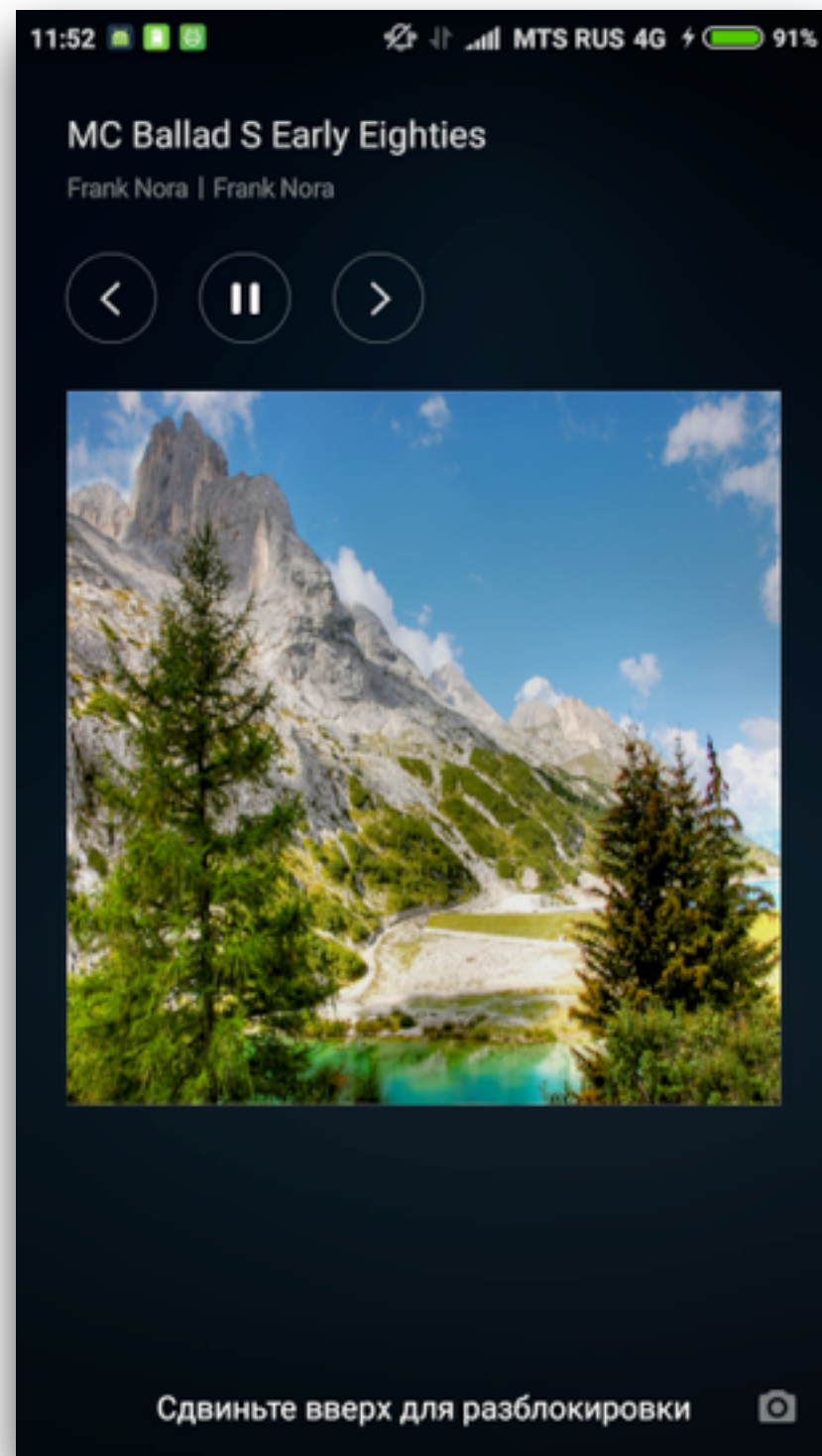
```
onAudioFocusChange(int focusChange)
```

AUDIOFOCUS_GAIN

AUDIOFOCUS_LOSS

AUDIOFOCUS_LOSS_TRANSIENT_CAN_DUCK





MIUI 8.2



onPlay	onPause	onStop	onSkip...
setActive(true)		setActive(false)	
setMetadata			setMetadata
setPlaybackState	setPlaybackState	setPlaybackState	
requestAudioFocus		abandonAudioFocus	



Notifications

Для API >= 21 обязательно!

```
case PlaybackStateCompat.STATE_PLAYING: {  
    startForeground(NOTIFICATION_ID,  
        getNotification(playbackState));  
    break;  
}  
case PlaybackStateCompat.STATE_PAUSED: {  
    NotificationManagerCompat  
        .from(PlayerService.this)  
        .notify(NOTIFICATION_ID,  
            getNotification(playbackState));  
    stopForeground(false);  
    break;  
}  
default: {  
    stopForeground(true);  
    break;  
}
```



Notifications

```
// MediaStyleHelper заполняет уведомление метаданными трека.  
// Хелпер любезно написал Ian Lake / Android Framework Developer  
// at Google и выложил здесь:  
// https://gist.github.com/ianhanniballake/47617ec3488e0257325c  
NotificationCompat.Builder builder = MediaStyleHelper  
    .from(this, mediaSession);  
  
builder.addAction(  
    new NotificationCompat.Action(  
        android.R.drawable.ic_media_previous,  
        getString(R.string.previous),  
        MediaButtonReceiver.buildMediaButtonPendingIntent(  
            this,  
            PlaybackStateCompat.ACTION_SKIP_TO_PREVIOUS)))  
;  
  
// Аналогично для  
// ACTION_SKIP_TO_NEXT и  
// ACTION_PLAY_PAUSE, ACTION_STOP
```



Notifications

```
builder.setStyle(new MediaStyle()  
    // В компактном варианте показывать Action с указанным  
    // порядковым номером  
    .setShowActionsInCompactView(1)  
    // Отображать крестик в углу уведомления для его закрытия.  
    // Это связано с тем, что в API < 21  
    // пользователь не мог смахнуть уведомление  
    // foreground-сервиса даже после stopForeground.  
    // На API >= 21 крестик не отображается,  
    // там просто смахиваем уведомление.  
    .setShowCancelButton(true)  
    // Указываем, что делать при нажатии на крестик или смахивании  
    .setCancelButtonIntent(  
        MediaButtonReceiver.buildMediaButtonPendingIntent(this,  
            PlaybackStateCompat.ACTION_STOP))  
    // Передаем токен. Это важно для Android Wear. Если токен не  
    // передать, кнопка на Android Wear будет отображаться,  
    // но не будет работать  
    .setMediaSession(mediaSession.getSessionToken()));
```



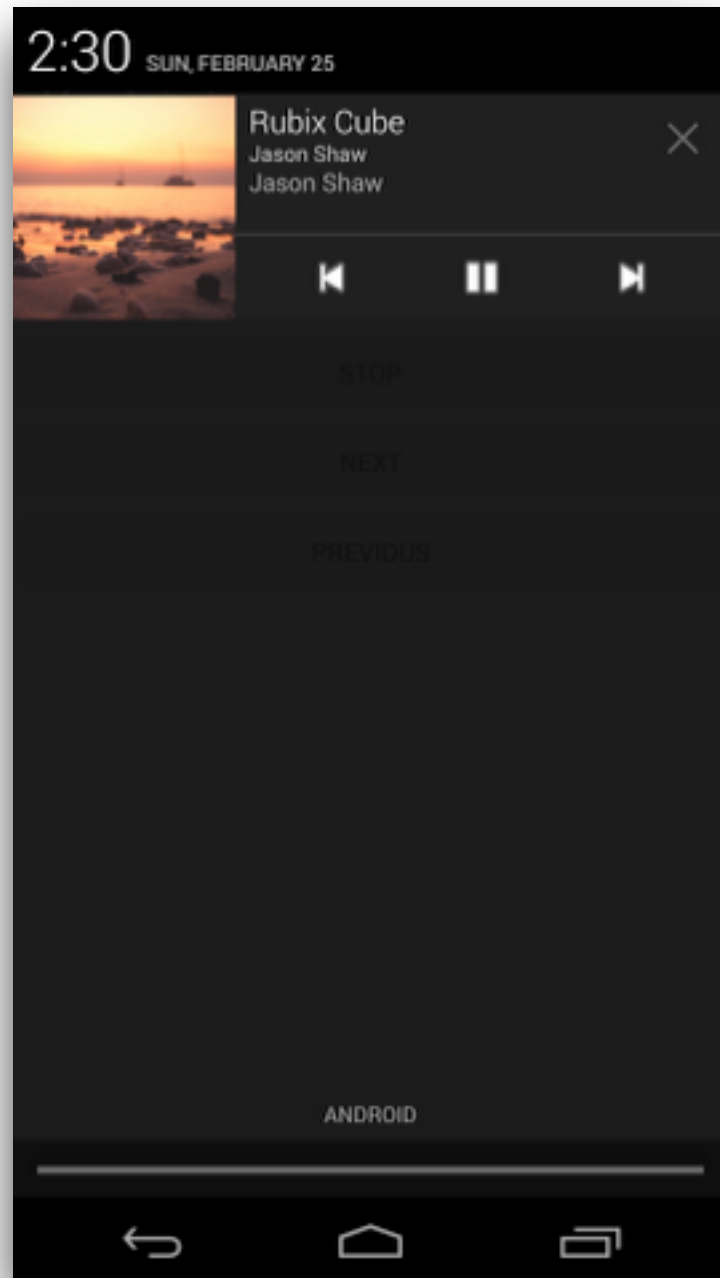
Notifications

// Не отображать время создания уведомления
`builder.setShowWhen(false);`

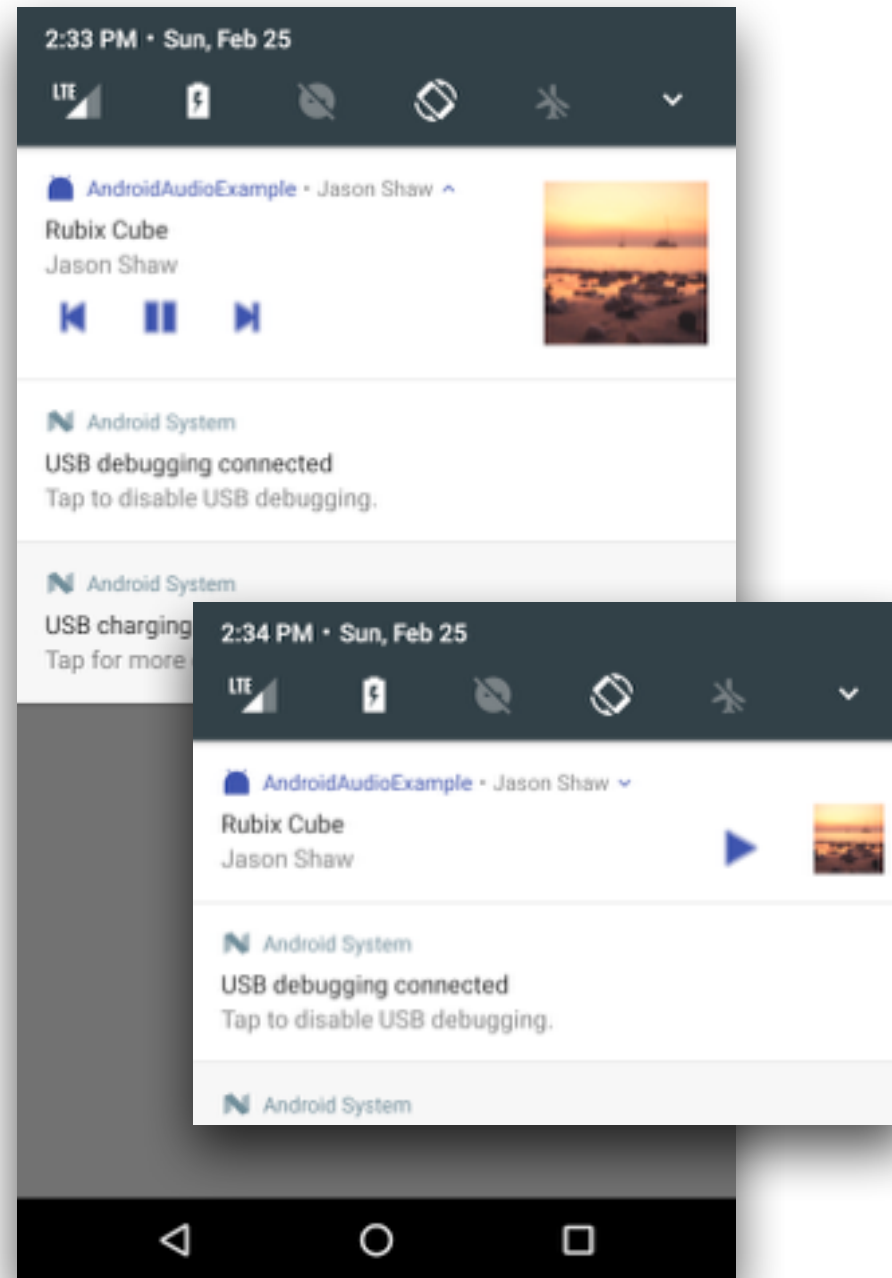
*// Магия! Без этой строки уведомления не отображаются
// на Android Wear и криво отображаются на самом телефоне*
`builder.setPriority(NotificationCompat.PRIORITY_HIGH);`

// Не надо каждый раз вываливать уведомление на пользователя
`builder.setOnlyAlertOnce(true);`

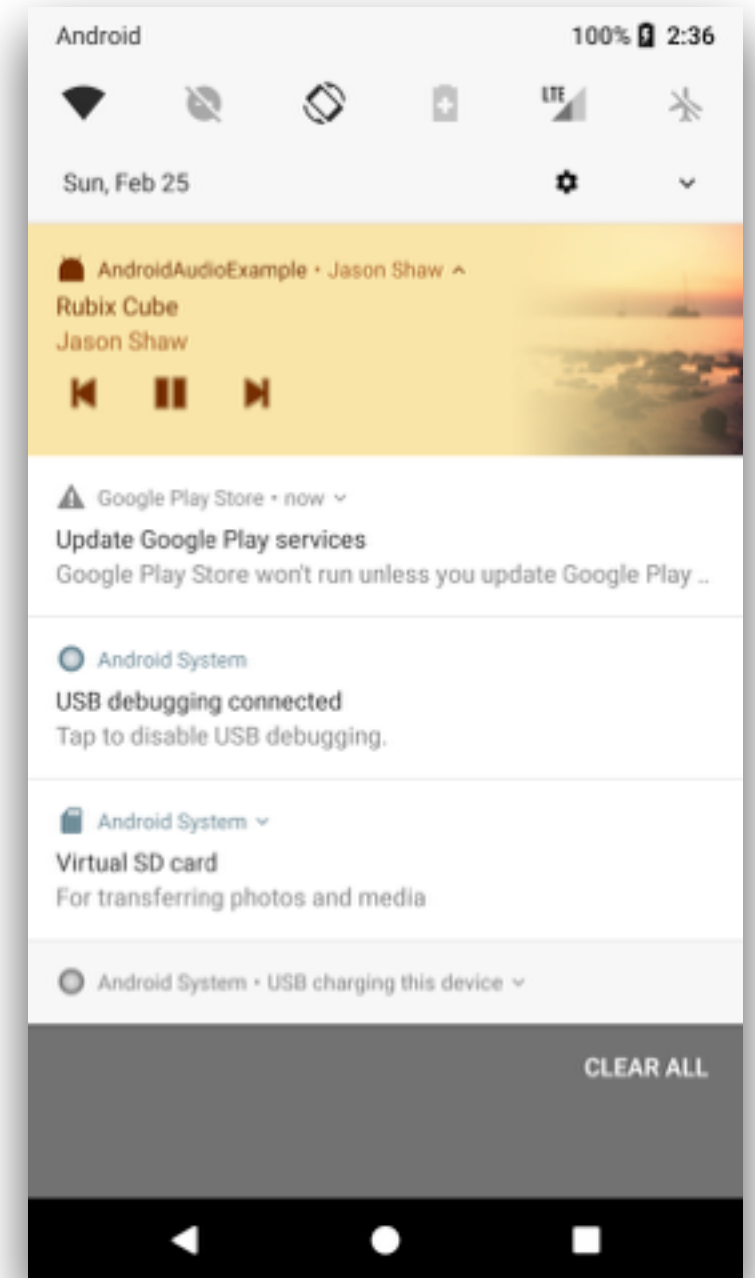




Android 4.4



Android 7.0



Android 8.0



onPlay	onPause	onStop	onSkip...
setActive(true)		setActive(false)	
setMetadata			setMetadata
setPlaybackState	setPlaybackState	setPlaybackState	
requestAudioFocus		abandonAudioFocus	
startForeground	stopForeground(false)	stopForeground(true)	



Started service

Мы не хотим, чтобы сервис убили

onPlay()

```
startService(new Intent(getApplicationContext(),  
    PlayerService.class));
```

onStop()

```
stopSelf()
```



ACTION_AUDIO_BECOMING_NOISY

```
BroadcastReceiver becomingNoisyReceiver = new BroadcastReceiver() {  
    @Override  
    public void onReceive(Context context, Intent intent) {  
        if (ACTION_AUDIO_BECOMING_NOISY.equals(intent.getAction())) {  
            mediaSessionCallback.onPause();  
        }  
    }  
};
```

[onPlay\(\)](#)

```
registerReceiver(becomingNoisyReceiver,  
    new IntentFilter(AudioManager.ACTION_AUDIO_BECOMING_NOISY));
```

[onPause\(\)](#)

```
unregisterReceiver(becomingNoisyReceiver);
```

[onStop\(\)](#)

```
unregisterReceiver(becomingNoisyReceiver);
```



onPlay	onPause	onStop	onSkip...
setActive(true)		setActive(false)	
setMetadata			setMetadata
setPlaybackState	setPlaybackState	setPlaybackState	
requestAudioFocus		abandonAudioFocus	
startForeground	stopForeground(false)	stopForeground(true)	
startService		stopSelf	
registerReceiver	unregisterReceiver	unregisterReceiver	



A few more things...



ExoPlayer (no MediaPlayer please)

<https://developer.android.com/guide/topics/media/exoplayer.html>

- ✗ Сложнее API
- ✗ Только с Android 4.1 (API 16)
- ✓ Официально рекомендовано Гуглом
- ✓ Доступен в исходниках
- ✓ Деплоймент вместе с приложением
- ✓ Расширяемость (кодеки/кеширование)
- ✓ Всякие вкусности “из коробки”
(потокковое аудио/видео, регулировка скорости воспроизведения)



MediaBrowserService

Публикация информации о доступных треках (например, для Android Auto)

<https://developer.android.com/reference/android/service/media/MediaBrowserService.html>

```
<service
  android:name=".service.PlayerService"
  android:exported="true">
  <intent-filter>
    <action android:name="android.media.browse.MediaBrowserService"/>
    <action android:name="android.intent.action.MEDIA_BUTTON" />
  </intent-filter>
</service>
```

- Наследуем наш сервис от MediaBrowserServiceCompat
- Реализуем абстрактные методы
 - [onGetRoot\(clientPackageName, ...\)](#)
 - [onLoadChildren\(parentId, ...\)](#)
- Сообщаем MediaBrowserService токен нашего MediaSession
 - [setSessionToken\(mediaSession.getSessionToken\(\)\)](#)
- Добавляем в манифест информацию о поддержке Android Auto



Links

- Статья на Хабре “Android и звук: как делать правильно”

<https://habrahabr.ru/post/339416/>

- Рабочий код на GitHub

<https://github.com/SergeyVinyar/AndroidAudioExample>

- Universal Android Music Player Sample by Google

<https://github.com/googlesamples/android-UniversalMediaPlayer>

- Media playback the right way (Big Android BBQ 2015) by Ian Lake

<https://www.youtube.com/watch?v=XQwe30cZffg>

