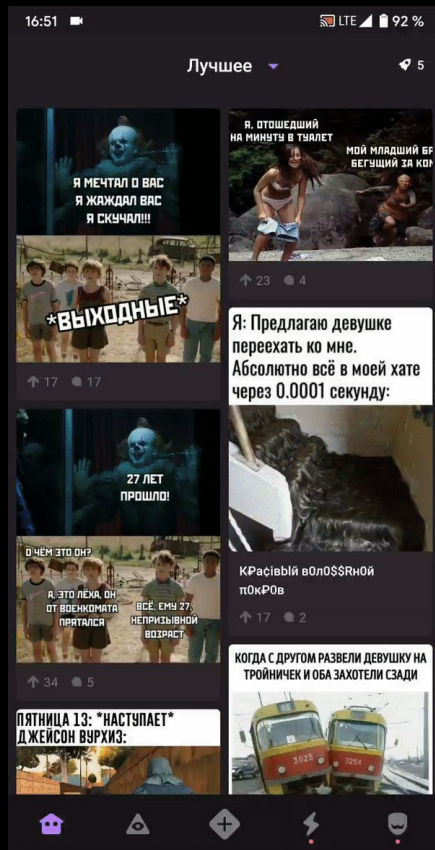




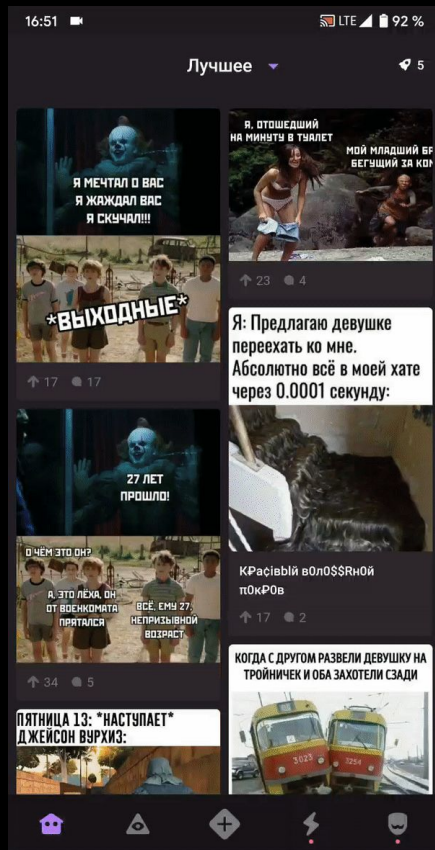
Михаил Зинченко

# Как сделать приложение с *ИЗЮМИНКОЙ*\*?

# Лента



# Лента

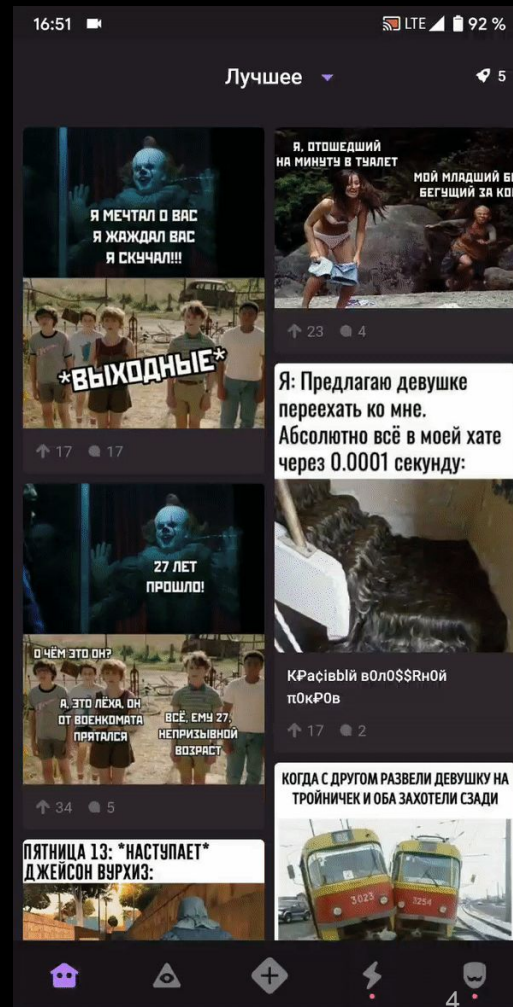


# Пост



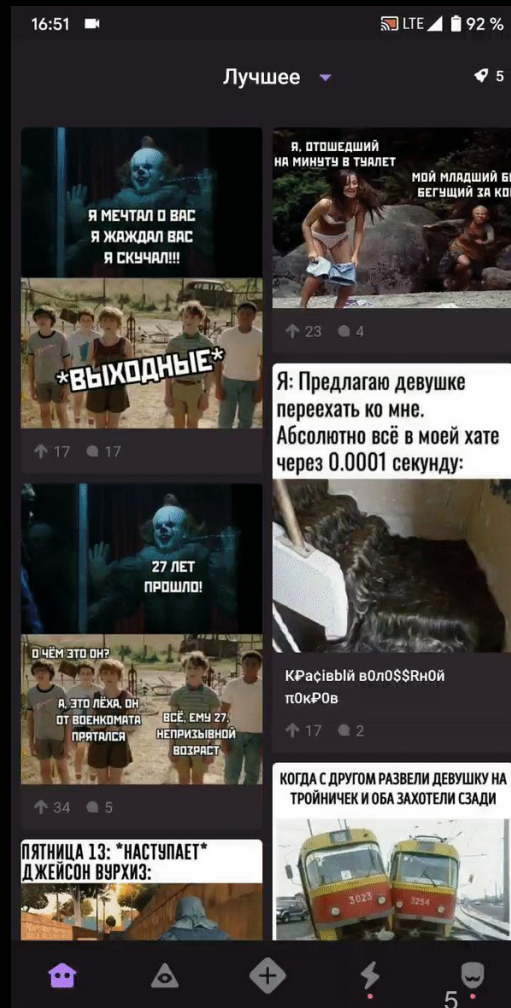
# Лента

- Множество лент



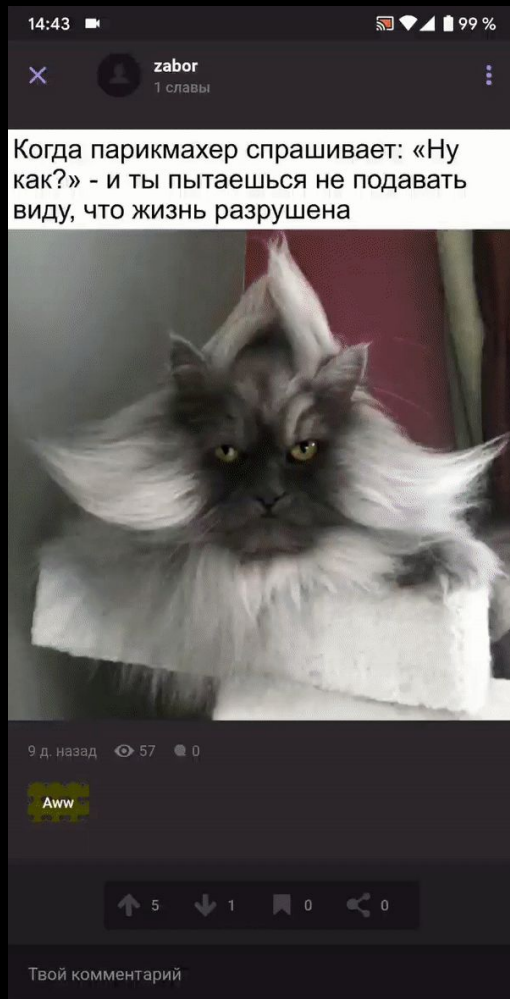
# Лента

- Множество лент
- Как синхронизировать посты в нескольких лентах?



# Синхронизация лент: Решение

Добавить сущность на уровне `interactor`, отвечающую за обработку и оповещение о событиях изменения поста.



# Свайп постов

- Горизонтальная лента внутри поста: что использовать: `RecyclerView` или `ViewPager`?



# RecyclerView

- + Нативная работа с динамическими списками
- + Огромный набор notify... методов адаптера
- + DiffUtil
- + Возможность шарить пул между ресайклерами
- + Возможность вручную задать количество подгружаемых элементов сзади и спереди
- Нужно писать кастомный SnapHelper
- Нельзя напрямую положить фрагмент

# RecyclerView

- + Нативная работа с динамическими списками
- + Огромный набор notify... методов адаптера
- + DiffUtil
- + Возможность шарить пул между ресайклерами
- + Возможность вручную задать количество подгружаемых элементов сзади и спереди
- ~~— Нужно писать кастомный SnapHelper~~
- ~~— Нельзя напрямую положить фрагмент~~
- + ViewPager2

# ViewPager

- + Простая навигация по элементам
  - + Поддержка фрагментов
  - + Можно легко встроить пагинацию (нет)
- 
- Только один notify... метод
  - Плохая масштабируемость

# Лента: увеличение сложности

- **Посты ленты**
- **Закрепленные посты**
- **Черновики**
  - **отображаются только в вертикальной ленте**
  - **отображаются только в ленте профиля**
- **Рекламные посты**
  - **отображаются только в горизонтальной ленте**
  - **подгружаются динамически**

# PaginationPagerAdapter

```
var items: List<T>
```

```
var isNextPageLoading: Boolean
```

```
var onLoadMoreCallback: () -> Unit
```

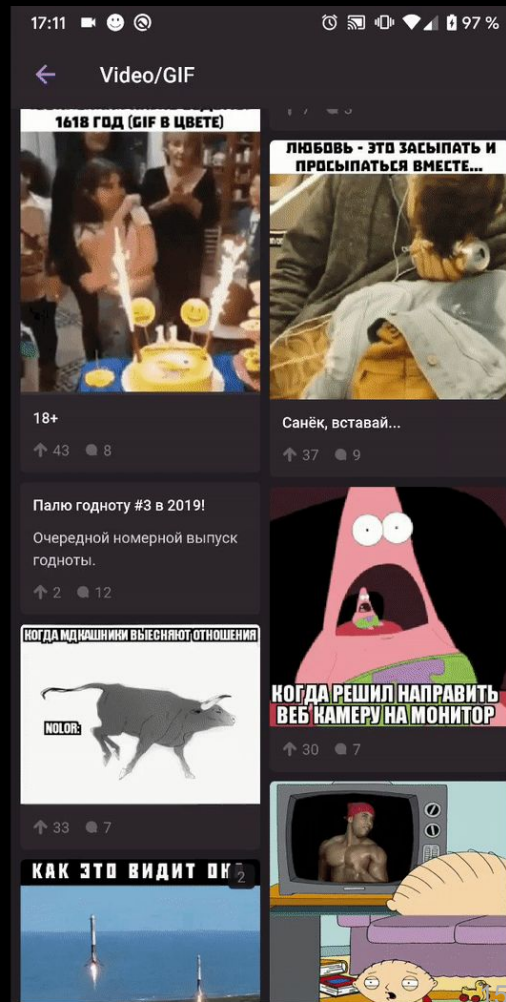
```
protected fun shouldLoadMore(position: Int): Boolean
```

```
protected open fun onPageSelectedAction(pos: Int) {  
    if (shouldLoadMore(pos)) onLoadMoreCallback()  
}
```



# Лента: плеер

- Как воспроизводить анимированный контент?



# Проблемы подхода с Gif

- Гифки очень много весят
- Сложная реализация потокового воспроизведения
- Количество фреймов на слабых девайсах оставляет желать лучшего



**Решение:**  
**Плеер с поддержкой**  
**mp4/webm**

# Лента: видеоплеер

- Реализация на основе: ExoPlayer2

```

com.google.android.exoplayer2.ExoPlaybackException: com.google.android.exoplayer2.mediacodec.MediaCodecRenderer$DecoderInitializationException: Decoder init failed: OMX.qcom.video.decoder.avc, Format(1, null, null, video/avc, null, -1.0], [-1, -1])
    at com.google.android.exoplayer2.mediacodec.MediaCodecRenderer.maybeInitCodec(MediaCodecRenderer.java:479)
    at com.google.android.exoplayer2.mediacodec.MediaCodecRenderer.reinitializeCodec(MediaCodecRenderer.java:1261)
    at com.google.android.exoplayer2.mediacodec.MediaCodecRenderer.onInputFormatChanged(MediaCodecRenderer.java:1111)
    at com.google.android.exoplayer2.video.MediaCodecVideoRenderer.onInputFormatChanged(MediaCodecVideoRenderer.java:552)
    at com.google.android.exoplayer2.mediacodec.MediaCodecRenderer.render(MediaCodecRenderer.java:647)
    at com.google.android.exoplayer2.ExoPlayerImplInternal.doSomeWork(ExoPlayerImplInternal.java:529)
    at com.google.android.exoplayer2.ExoPlayerImplInternal.handleMessage(ExoPlayerImplInternal.java:300)
    at android.os.Handler.dispatchMessage(Handler.java:103)
    at android.os.Looper.loop(Looper.java:214)
    at android.os.HandlerThread.run(HandlerThread.java:67)
Caused by: com.google.android.exoplayer2.mediacodec.MediaCodecRenderer$DecoderInitializationException: Decoder init failed: OMX.qcom.video.decoder.avc, Format(1, null, null, video/avc, null, -1.0], [-1, -1])
    at com.google.android.exoplayer2.mediacodec.MediaCodecRenderer.initCodecWithFallback(MediaCodecRenderer.java:753)
    at com.google.android.exoplayer2.mediacodec.MediaCodecRenderer.maybeInitCodec(MediaCodecRenderer.java:474) <9 more...>
Caused by: android.media.MediaCodec$CodeException: Failed to initialize OMX.qcom.video.decoder.avc, error 0xffffffff
    at android.media.MediaCodec.native_setup(Native Method)
    at android.media.MediaCodec.<init>(MediaCodec.java:1853)
    at android.media.MediaCodec.createByCodecName(MediaCodec.java:1831)
    at com.google.android.exoplayer2.mediacodec.MediaCodecRenderer.initCodec(MediaCodecRenderer.java:802)
    at com.google.android.exoplayer2.mediacodec.MediaCodecRenderer.initCodecWithFallback(MediaCodecRenderer.java:745) <10 more...>
2019-06-13 10:32:27.709 809-809/? I/Zygote: Process 23586 exited due to signal 9 (Killed)
2019-06-13 10:32:27.769 1270-3967/? I/ActivityManager: Process com.google.android.play.games.ui (pid 23586) has died: cch+55 CEM
2019-06-13 10:32:27.769 1270-1315/? I/libprocessgroup: Successfully killed process cgroup uid 10034 pid 23586 in 0ms
2019-06-13 10:32:28.571 23227-26510/is.mdk.app.debug I/OMXClient: IOmx service obtained

```

# Лента: видеоплеер

- Реализация на основе: ExoPlayer2
- Для Player и PlayerView нужен свой жизненный цикл и пул

```
interface PlayableHolder {  
    fun init()  
    fun stop(isHard: Boolean = false)  
    fun getPlayerTag(): String  
}
```

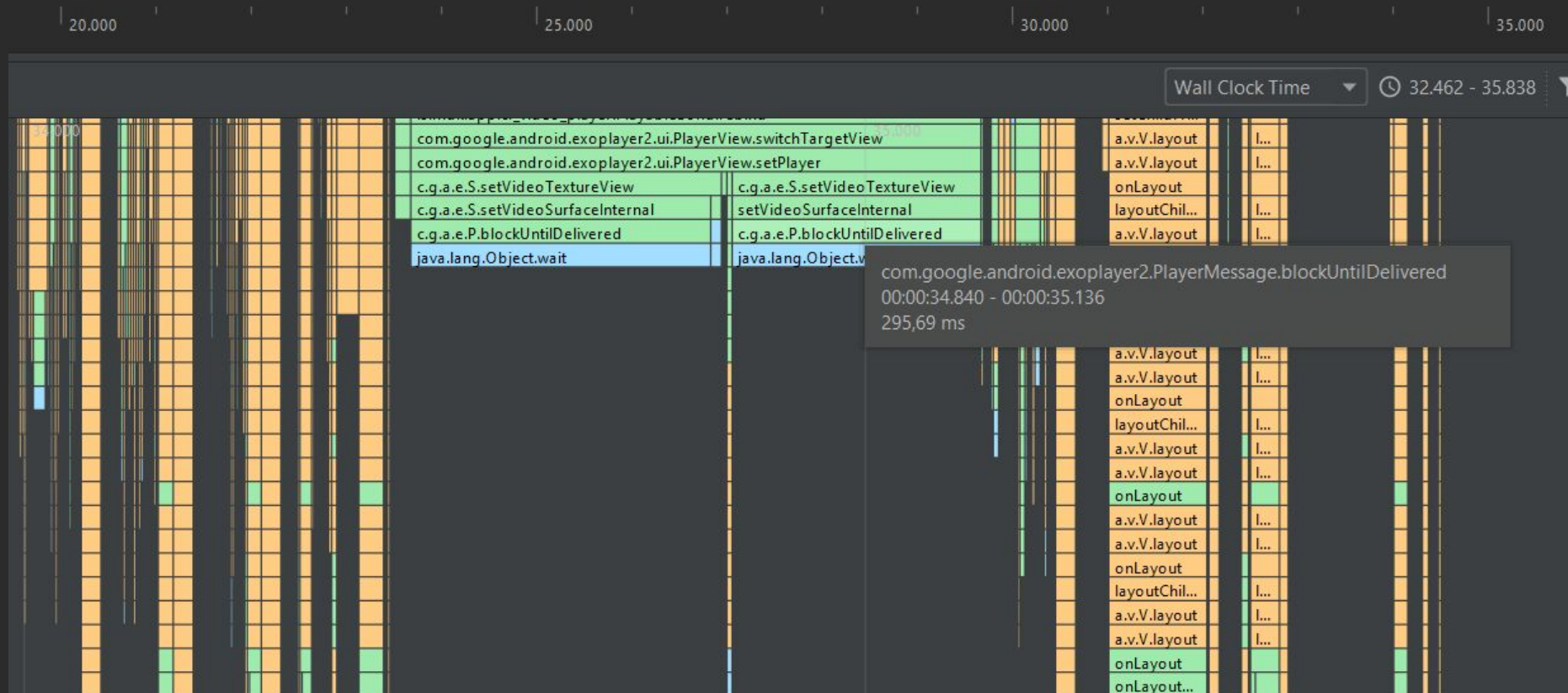
```
recyclerView?.addOnScrollListener {  
    //debounce  
    val visiblePositions = findVisibleRange(recyclerView)  
    changePlayingPositions(visiblePositions)  
}
```

```
class PlayerManager {  
    fun stop(tag: String, isHard: Boolean) {  
        val player = pool.getPlayer(tag)  
        if (player != null && !player.isStopped) {  
            player.playWhenReady = false  
            if (isHard) {  
                player.stop(true)  
            }  
        }  
    }  
}
```

```
class PlayerManager {  
    fun stop(tag: String, isHard: Boolean) {  
        val player = pool.getPlayer(tag)  
        if (player != null && !player.isStopped) {  
            player.playWhenReady = false  
            if (isHard) {  
                player.stop(true)  
            }  
        }  
    }  
}
```



```
class PlayerManager {  
  
    fun stop(tag: String, isHard: Boolean) {...}  
  
    fun init(  
        tag: String,  
        mediaUrl: String,  
        playerView: PlayerView  
    ) {  
        val player = pool.obtainPlayer(tag)  
        if (player.isStopped) { player.prepare(...) }  
        playerView.player = player  
        player.playWhenReady = true  
    }  
}
```



```
class PlayerManager {  
  
    fun stop(tag: String, isHard: Boolean) {...}  
  
    fun init(  
        tag: String,  
        mediaUrl: String,  
        context: Context  
    ): PlayerView {  
        val playable = pool.obtainPlayable(tag)  
        val (player, view) = playable  
        if (player.isStopped) { player.prepare(...) }  
        player.playWhenReady = true  
        return view  
    }  
}
```

```
class PlayablePool @Inject constructor() {  
  
    private val pool: Map<String, Playable>  
    var size = 4  
    val isFull: Boolean  
  
    fun getPlayables(): Collection<Playable>  
    fun getPlayable(tag: String): Playable? = pool[tag]  
  
    fun obtainPlayer(  
        tag: String,  
        context: Context  
    ): Playable? =  
        getPlayerWithTag(tag) ?:  
        getIdlePlayer()?.also { changeTag(tag, it) } ?:  
        tryCreatePlayer(context, tag)
```

# Лента: видеоплеер

- Реализация на основе: ExoPlayer2
- Для Player и PlayerView нужны свой жизненный цикл и пул
- Необходимо ручное измерение производительности девайса

# Performance module

`PerformanceAnalyzer.analyzePerformance()`

# Performance module

```
PerformanceAnalyzer.analyzePerformance()
```

```
PerformanceRate { LOW, MEDIUM, HIGH }
```

# Performance module

`PerformanceAnalyzer.analyzePerformance()`

`PerformanceRate { LOW, MEDIUM, HIGH }`

`CPU, MEMORY, SYSTEM`



# Performance module

```
PerformanceAnalyzer.analyzePerformance()
```

```
PerformanceRate { LOW, MEDIUM, HIGH }
```

```
CPU, MEMORY, SYSTEM
```

```
if (isMediatek()) PerformanceRate.LOW
```

# Performance strategy

```
interface PlayerPerformanceStrategy {  
    val playersCount: Int  
    val isPlayingOnlyOnCompletelyVisibleViews: Boolean  
}
```

# Performance strategy

```
interface PlayerPerformanceStrategy
```

```
class FeedLowPerformanceStrategy (2, true)
```

```
class FeedHighPerformanceStrategy(4, false)
```

```
class PostLowPerformanceStrategy (2, false)
```

```
class PostHighPerformanceStrategy (4, false)
```

# Лента: выводы

- Для горизонтальной ленты используем RecyclerView/ViewPager 2

# Лента: выводы

- Для горизонтальной ленты используем RecyclerView/ViewPager 2
- Для шаринга лент лучше использовать одну точку доступа

# Лента: выводы

- Для горизонтальной ленты используем `RecyclerView/ViewPager 2`
- Для шаринга лент лучше использовать одну точку доступа
- Для анимированного контента следует использовать видео-форматы

# Лента: выводы

- Для горизонтальной ленты используем `RecyclerView/ViewPager 2`
- Для шаринга лент лучше использовать одну точку доступа
- Для анимированного контента следует использовать видео-форматы
- При работе с видео на Android готовьтесь к головной боли =)

# Пост

- Много типов контента, расположенных произвольным образом





# EasyAdapter

```
val easyAdapter = EasyAdapter()  
...  
rv.adapter = easyAdapter  
...  
adapter.setItems(createPostItems(...))
```

[GitHub](#)

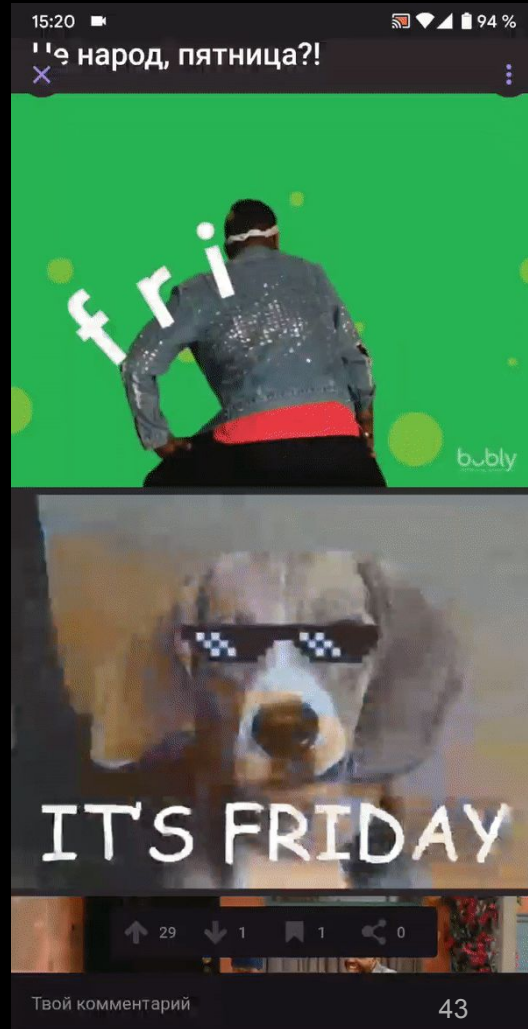
```
add(post.author, authorController)
addIf(post.hasTitle, post.title, titleController)

var prevItem: MediaItem? = null
post.mediaItems.forEach { mediaItem ->
    if (mediaItem.type != mediaItemPrev?.type)
        add(separatorController)

    when (mediaItem.type) {
        MediaType.TEXT_ONLY -> add(mediaItem, textCnt)
        MediaType.IMAGE, MediaType.GIF -> add(mediaItem, imageCnt)
        MediaType.VIDEO -> add(mediaItem, videoCnt)
        MediaType.LINK -> add(mediaItem, previewCnt)
        MediaType.UNKNOWN -> {}
    }
    prevItem = mediaItem
}
add(post, infoController)
```

# Пост

- Много типов контента, расположенных произвольным образом
- Пост и комментарии идут одним большим списком



# AsyncInflater

```
val context = requireContext()
val mainFrame = FrameLayout(context)
mainFrame.id = R.id.horizontal_feed_root_container
val asyncInflater = AsyncLayoutInflater(context)
asyncInflater.inflate(
    R.layout.fragment_post,
    mainFrame,
    { child, _, parent ->
        parent?.addView(child)
        initViews()
        presenter.onFinishInflate()
    }
)
return mainFrame
```

# Анимация

- SharedElementTransition



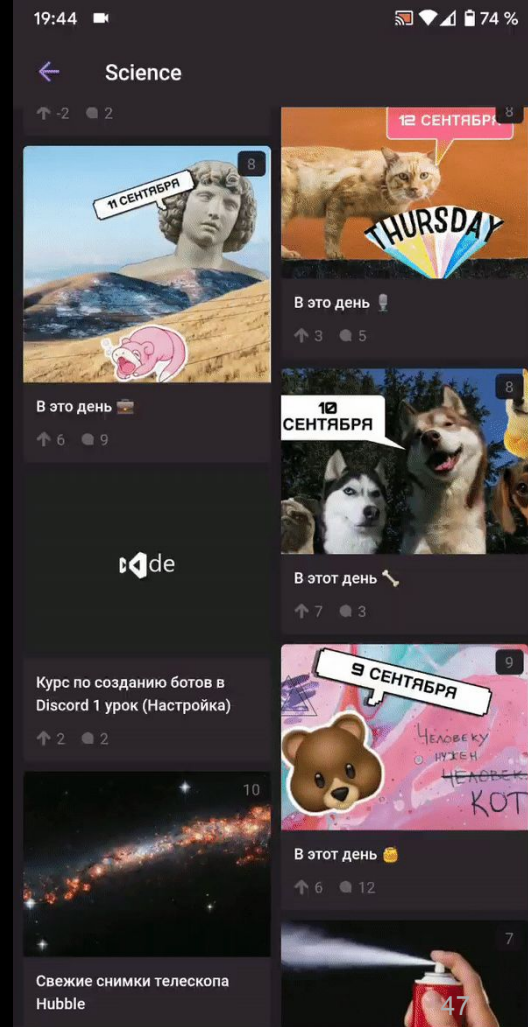
# Анимация

- **SharedElementTransition**
  - Сложный механизм
  - Ожидание окончания `inflate`
  - **Shared-элемент** может не поместиться на экран, и это будет выглядеть плохо



# Анимация

- **SharedElementTransition**
- **Custom Circular Reveal Animation**
  - + Простая реализация
  - + Можно запустить сразу же, анимация не замораживает экран
  - + Контекст поста так же хорошо читается
  - + Прикольно выглядит вместе с Ripple



# Пост

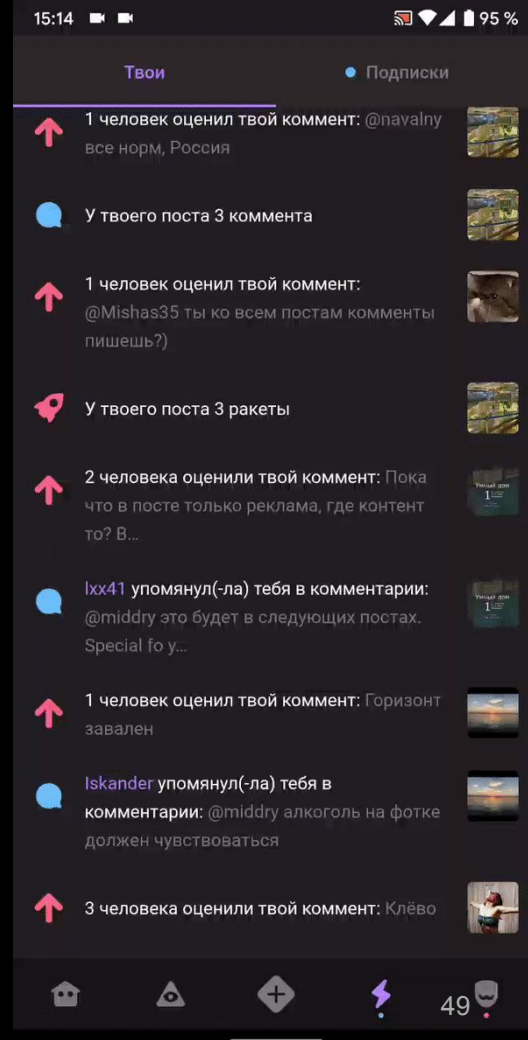
- Много типов контента, расположенных произвольным образом
- Пост и комментарии идут одним большим списком
- Пагинация комментариев в обе стороны

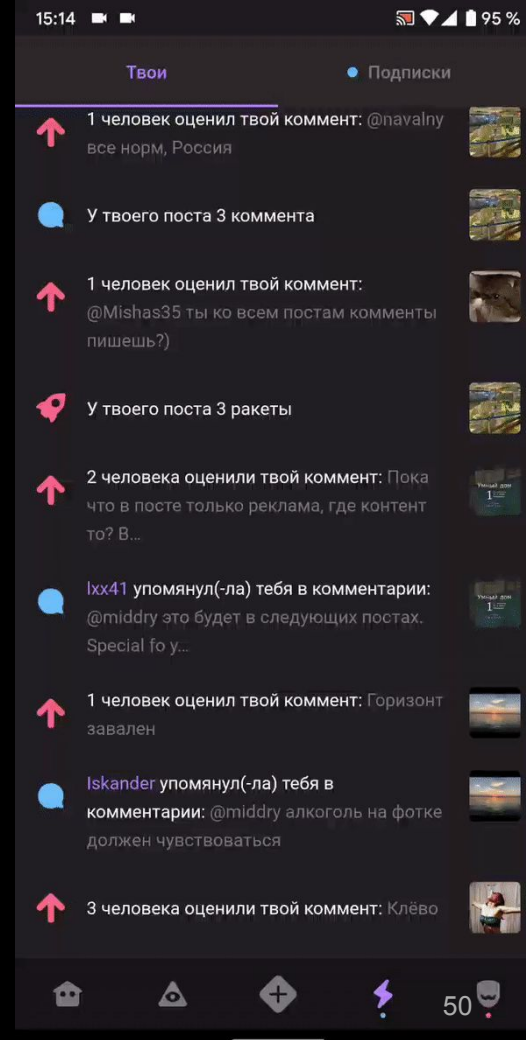




# Пост

- Много типов контента, расположенных произвольным образом
- Пост и комментарии идут одним большим списком
- Пагинация комментариев в обе стороны
- Возможность перейти к любому комментарию



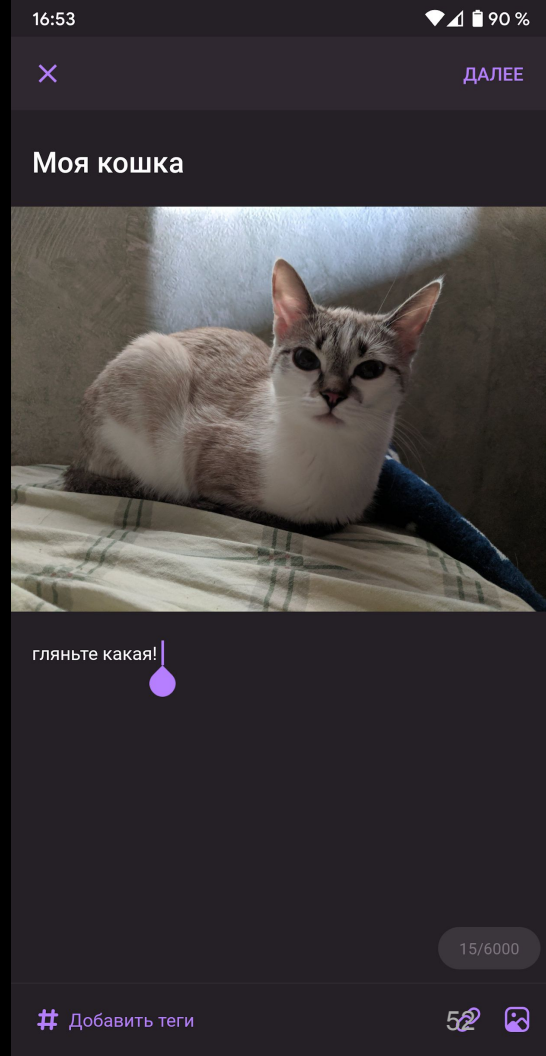


# Открытие комментария

1. Передаем id комментария при старте экрана
2. Добавляем полноэкранный ладер
3. Запрашиваем пачку комментариев, в которой находится открываемый.
4. Отрисовываем пришедшие комментарии
5. Выжидаем, как только произойдет отрисовка элементов RecyclerView и скроллим к открытому комментарию.
6. Скрываем ладер

# Создание поста

- Один большой RecyclerView
- Каждые 30 секунд сохраняется локальная копия контента
- Текст автоматически разделяется на абзацы, есть лимит ввода
- Элементы можно перетаскивать и менять местами



# Drag and Drop в RecyclerView





All

Images

Videos

News

Maps

More

Settings

Tools

About 23,100 results (0.46 seconds)

### Drag and drop items in RecyclerView with GridLayoutManager - Stack ...

<https://stackoverflow.com/.../drag-and-drop-items-in-recyclerview-with-gridlayoutma...>

3 answers

There is actually a better way to achieve this. You can use some of the **RecyclerView** 's "companion" classes: **ItemTouchHelper** , which is. a utility class to add ...

#### People also search for

[itemtouchhelper drag and drop](#)[notifyitemmoved not working](#)[recyclerview ondraglistener](#)[itemtouchhelper remove](#)[xamarin android recyclerview drag](#)[android recyclerview drag and drop library](#)

### Drag and Drop in android using RecyclerView - Stack Overflow

<https://stackoverflow.com/questions/.../drag-and-drop-in-android-using-recyclerview>

1 answer

May 3, 2017 - Callback() { int dragFromPosition = -1; int dragToPosition = -1; @Override public int getMovementFlags(RecyclerView recyclerView, ...

### RecyclerView Drag and Drop like - FILE & FOLDER - Stack Overflow

<https://stackoverflow.com/questions/.../recyclerview-drag-and-drop-like-file-folder>

Maybe this will help you: **Drag and drop** items in **RecyclerView** with **GridLayoutManager**. More related tutorial: **Drag and Swipe** with **RecyclerView**.

### Drag and drop in between items of recycler view - Stack Overflow

<https://stackoverflow.com/questions/.../drag-and-drop-in-between-items-of-recycler-vi...>

May 3, 2017 - Now I want to implement **drag and drop** functionality. Any item can be dragged anywhere i.e. it can be replaced with other item or may be it can ...

### Drag and Drop between two RecyclerView - Stack Overflow

<https://stackoverflow.com/questions/.../drag-and-drop-between-two-recyclerview>

4 answers

It is a very basic code that matches your requirement - **Drag and drop** within a single **recyclerview** as well as between two different **recyclerviews**. Hope this ...

### 3 Answers

active

oldest

votes

Не нашли ответ? [Задайте вопрос на Stack Overflow на русском.](#)



97



There is actually a better way to achieve this. You can use some of the `RecyclerView`'s "companion" classes:

[ItemTouchHelper](#), which is

a utility class to add swipe to dismiss and drag & drop support to `RecyclerView`.

and its [ItemTouchHelper.Callback](#), which is

the contract between `ItemTouchHelper` and your application

```
// Create an `ItemTouchHelper` and attach it to the `RecyclerView`
ItemTouchHelper ith = new ItemTouchHelper(_ithCallback);
ith.attachToRecyclerView(rv);
```

▲  
0 I solved same problem by ItemTouchHelper. Its little but tricky but simple after understand. Suppose you have a recycleView called mRecyclerView.

```
public void initRecyclerView(){  
    //initiate your recycleView here ....
```





0

I solved same problem by ItemTouchHelper. Its little but tricky but simple after understand. Suppose you have a recycleView called mRecyclerView.

```
public void initRecyclerView(){  
    //initiate your recycleView here ....
```

# Drag and Swipe with RecyclerView

## Part One: Basic ItemTouchHelper Example



Paul Burke

Follow

Jun 23, 2015 · 5 min read

▲  
0  
I solved same problem by ItemTouchHelper. Its little but tricky but simple after understand. Suppose you have a recycleView called mRecyclerView.

```
public void initRecyclerView(){  
    //initiate your recycleView here ....
```

## Drag and Swipe with RecyclerView

### Part One: Basic ItemTouchHelper Example



Paul Burke [Follow](#)

Jun 23, 2015 · 5 min read

## RecyclerView Drag and Drop

Drag and Drop can be added in a RecyclerView using the ItemTouchHelper utility class.

Following are the important methods in the `ItemTouchHelper.Callback` interface which needs to be implemented:

▲ 0 I solved same problem by ItemTouchHelper. Its little but tricky but simple after understand. Suppose you have a recycleView called mRecyclerView.

```
public void initRecyclerView(){  
    //initiate your recycleView here ...
```

# Drag and Swipe with RecyclerView

## Part One: Basic ItemTouchHelper Example



Paul Burke [Follow](#)

Jun 23, 2015 · 5 min read

## RecyclerView Drag and Drop

Drag and Drop can be added in a RecyclerView using the ItemTouchHelper utility class.

Following are the important methods in the `ItemTouchHelper.Callback` interface which needs to be implemented:

▲ You have to implement `ItemTouchHelper` and this is an example about how to do it:

I solved same problem by **ItemTouchHelper**. Its little but tricky but simple after understand. Suppose you have a recycleView called mRecyclerView.

```
public void initRecyclerView(){  
    //initiate your recycleView here ...
```

## Drag and Swipe with RecyclerView

### Part One: Basic **ItemTouchHelper** Example



Paul Burke [Follow](#)

Jun 23, 2015 · 5 min read

## RecyclerView Drag and Drop

Drag and Drop can be added in a RecyclerView using the ItemTouchHelper utility class.

Following are the important methods in the **ItemTouchHelper.Callback** interface which needs to be implemented:

You have to implement **ItemTouchHelper** and this is an example about how to do it:

# ItemTouchHelper

+ Нативный подход

# ItemTouchHelper

- + **Нативный подход**
- + **Простота использования**

# ItemTouchHelper

- + **Нативный подход**
- + **Простота использования (ага)**

# ItemTouchHelper

- + **Нативный подход**
- + **Простота использования (ага)**
- + **Масштабируемость**

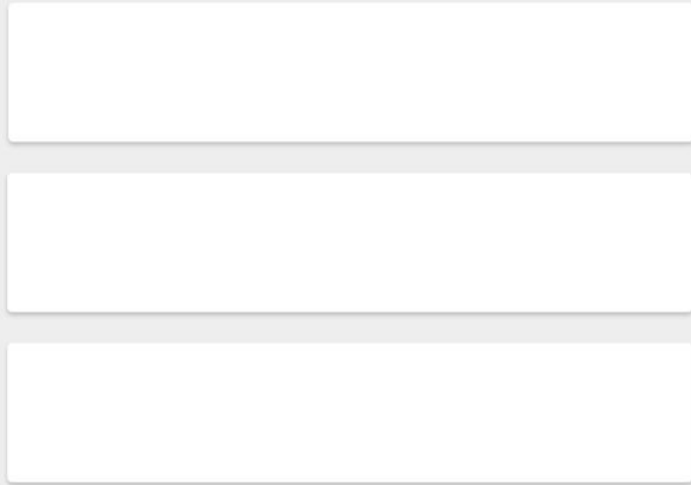


# ItemTouchHelper

- + **Нативный подход**
- + **Простота использования (ага)**
- + **Масштабируемость (ага)**

# ItemTouchHelper

- + Нативный подход
  - + Простота использования (ага)
  - + Масштабируемость (ага)
- 
- Перемещается сам View



# ItemTouchHelper

- + Нативный подход
  - + Простота использования (ага)
  - + Масштабируемость (ага)
- 
- Перемещается сам View
  - Подходит только для элементов с маленькой высотой

# ItemTouchHelper

- + **Нативный подход**
  - + **Простота использования (ага)**
  - + **Масштабируемость (ага)**
- 
- **Перемещается сам View**
  - **Подходит только для элементов с маленькой высотой**
  - **Необходимость ресайзить элементы с большой высотой**

# ItemTouchHelper

- + Нативный подход
  - + Простота использования (ага)
  - + Масштабируемость (ага)
- 
- Перемещается сам View
  - Подходит только для элементов с маленькой высотой
  - Необходимость ресайзить элементы с большой высотой
  - Много багов из-за ресайза
  - Перетаскивать элементы с большой высотой все равно неудобно



ЗАПОСТИТЬ



Slav Chen



# Добавить теги



# View.OnDragListener

- + Не перетаскивают сам элемент, перетаскивают скриншот
  - ⇒ Можно изменить представление этого скриншота как угодно, View не изменится



# View.OnDragListener

- + Не перетаскивают сам элемент, перетаскивают скриншот
  - ⇒ Можно изменить представление этого скриншота как угодно, View не изменится
- + Простой интерфейс

# View.OnDragListener

- + Не перетаскивают сам элемент, перетаскивают скриншот
  - ⇒ Можно изменить представление этого скриншота как угодно, View не изменится
- + Простой интерфейс
- + Широкий простор для кастомизации

# View.OnDragListener

- Не скроллится, когда доходит до конца экрана

# View.OnDragListener

- ~~Не скроллится, когда доходит до конца экрана~~ Можно написать кастомный скролл

# View.OnDragListener

- ~~Не скроллится, когда доходит до конца экрана~~ Можно написать кастомный скролл
- Придется писать много кастомной логики

# View.OnDragListener

- ~~Не скроллится, когда доходит до конца экрана~~ Можно написать кастомный скролл
- ~~Придется писать много кастомной логики~~ Все еще лучше, чем фиксить бесконечные баги

# View.OnDragListener

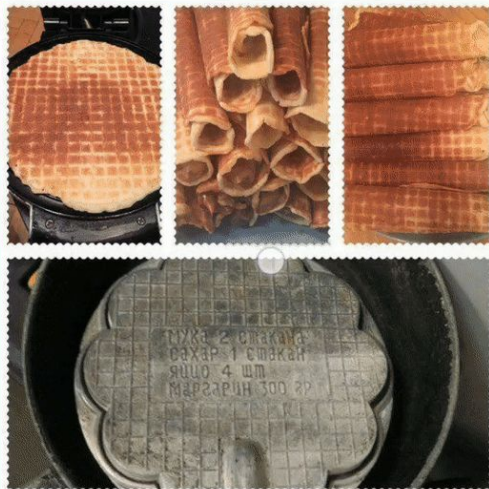
- ~~Не скроллится, когда доходит до конца экрана~~ Можно написать кастомный скролл
- ~~Придется писать много кастомной логики~~ Все еще лучше, чем фиксить бесконечные баги
- Мы можем переместить Shadow за Recycler или даже за экран приложения



POST

Title

Text



# Add tags





# View в неактивной области

```
onDragEnd = { _, _, _ -> cancelDrag() }
```

# View.OnDragListener

- ~~Не скроллится, когда доходит до конца экрана~~ Можно написать кастомный скролл
- ~~Придется писать много кастомной логики~~ Все еще лучше, чем фиксить бесконечные баги
- Мы можем переместить Shadow за Recycler или даже за экран приложения
- EditText ловит DragEvent как текст

# View.OnDragListener

- ~~Не скроллится, когда доходит до конца экрана~~ Можно написать кастомный скролл
- ~~Придется писать много кастомной логики~~ Все еще лучше, чем фиксить бесконечные баги
- Мы можем переместить Shadow за Recycler или даже за экран приложения
- EditText ловит DragEvent как текст (ну это вообще пушка)



POST

Title

привет, как дела?



# Add tags

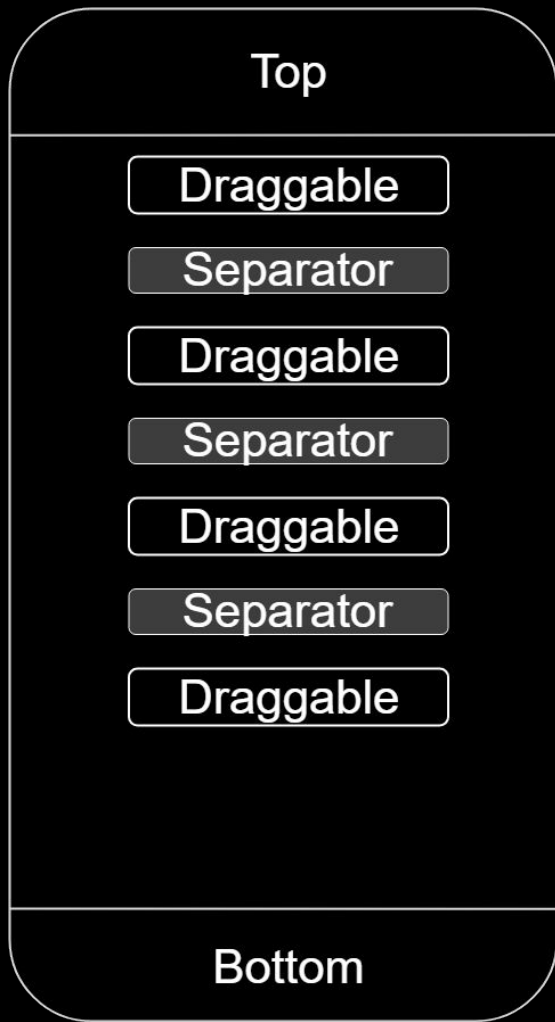


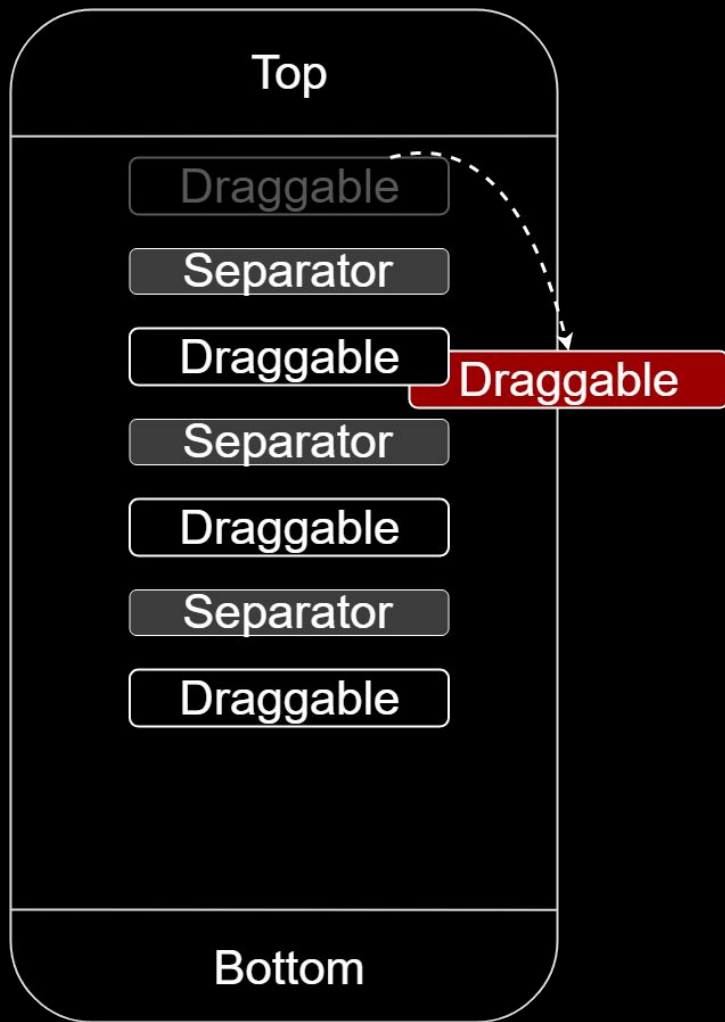
# EditText, глотающий DragEvent

```
@Override
public boolean onDragEvent(DragEvent event) {
    switch (event.getAction()) {
        ...
        case DragEvent.ACTION_DRAG_LOCATION:
            if (mText instanceof Spannable) {
                final int offset = ...
                Selection.setSelection(mSpannable, offset);
            }
            return true;
        case DragEvent.ACTION_DROP:
            if (mEditor != null) mEditor.onDrop(event);
            return true;
    }
}
```

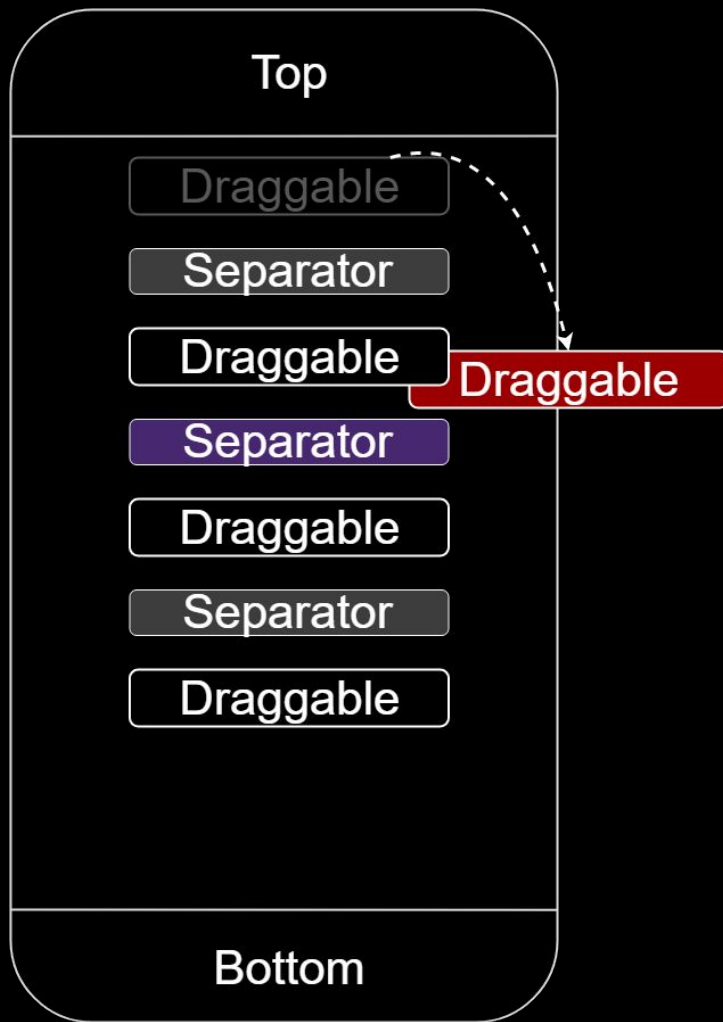
# EditText, глотающий DragEvent

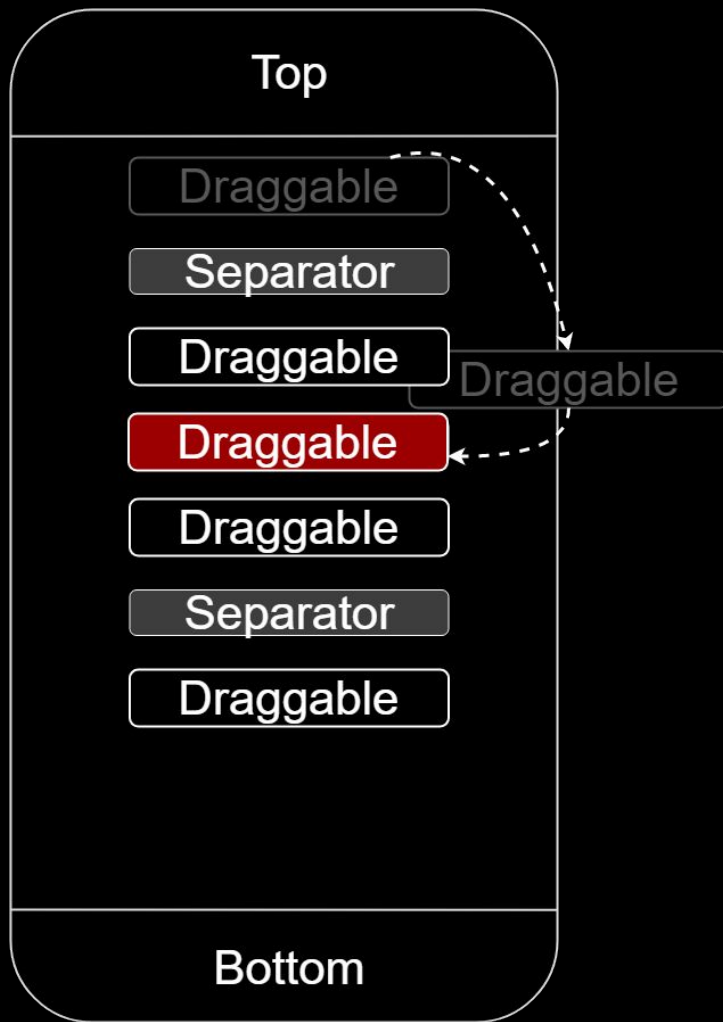
```
open class DraggableEditText @JvmOverloads constructor(  
    context: Context,  
    attrs: AttributeSet? = null  
) : EditText(context, attrs) {  
  
    override fun onDragEvent(event: DragEvent): Boolean {  
        return false  
    }  
}
```

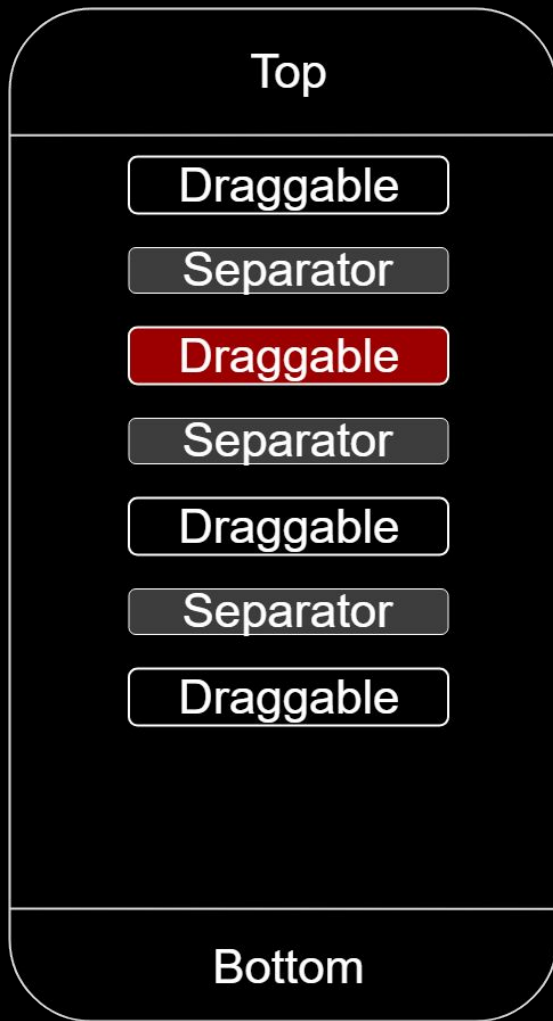














ДАЛЕЕ

## Van Gogh Museum



# Добавить теги



# Выводы по Drag'N'Drop

- Для невысоких однотипных элементов - `ItemTouchHelper`
- Для остального - кастомный `View.OnDragListener`
- Sample: [RecyclerDragHelper](#)

# Пробежимся по темам

- Свайп постов и синхронизация данных в лентах
- Видеоплеер в динамических лентах
- Рендеринг сложного нагруженного экрана поста
- Анимация открытия экрана поста
- Перетаскивание элементов в RecyclerView

# Q&A



- Telegram: @midery
- VK: [vk.com/midery](https://vk.com/midery)
- Email: midery@yandex.com

