

Android UI 进阶

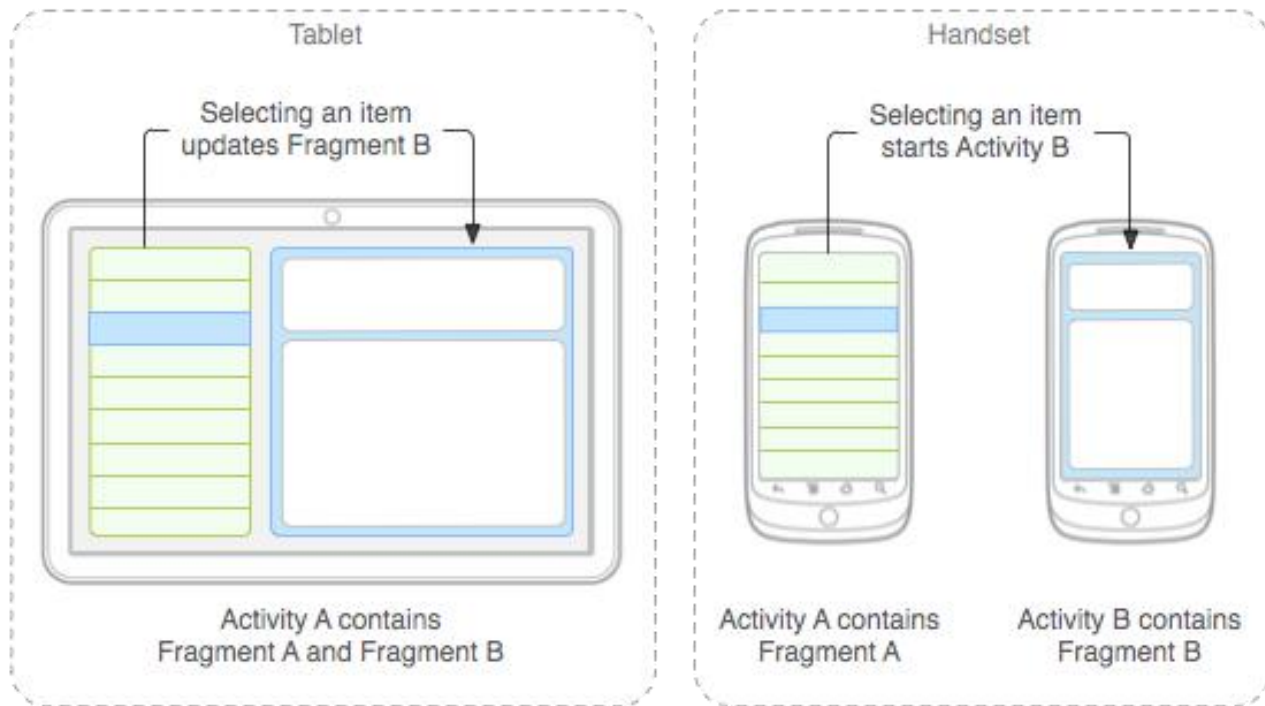
焦鹏文 | 字节跳动 Android 工程师
jiaopengwen@bytedance.com



Fragment

- 
- ❑ Understanding Fragments
 - ❑ Fragment Lifecycle
 - ❑ Embedding a Fragment Dynamically
 - ❑ Communicating with Activity

Understanding Fragments

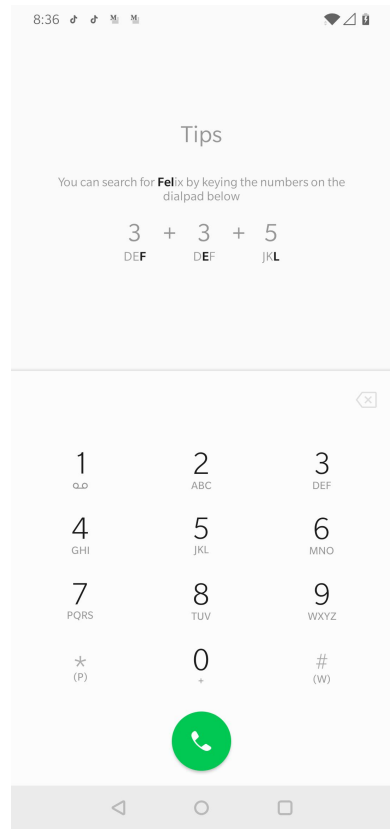
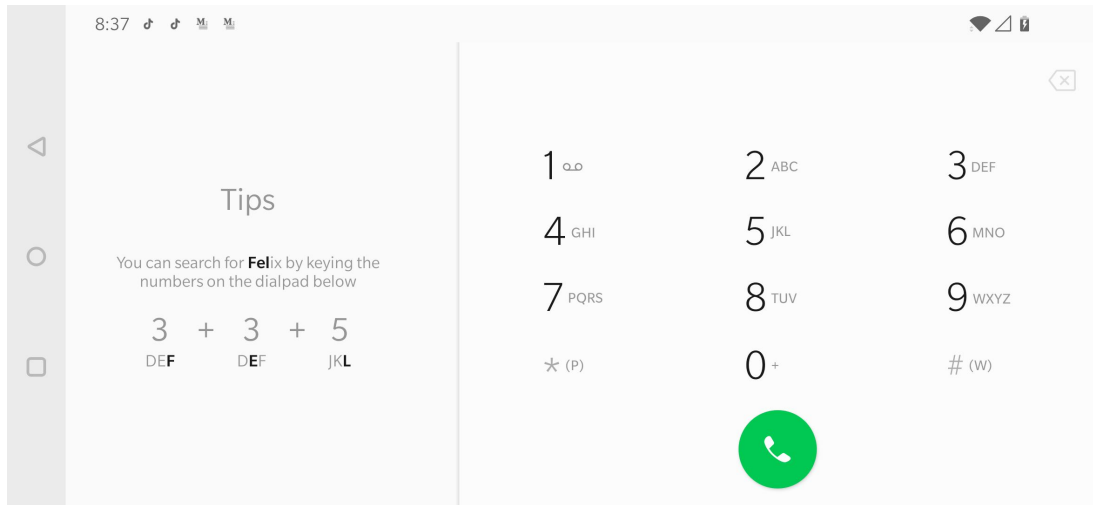




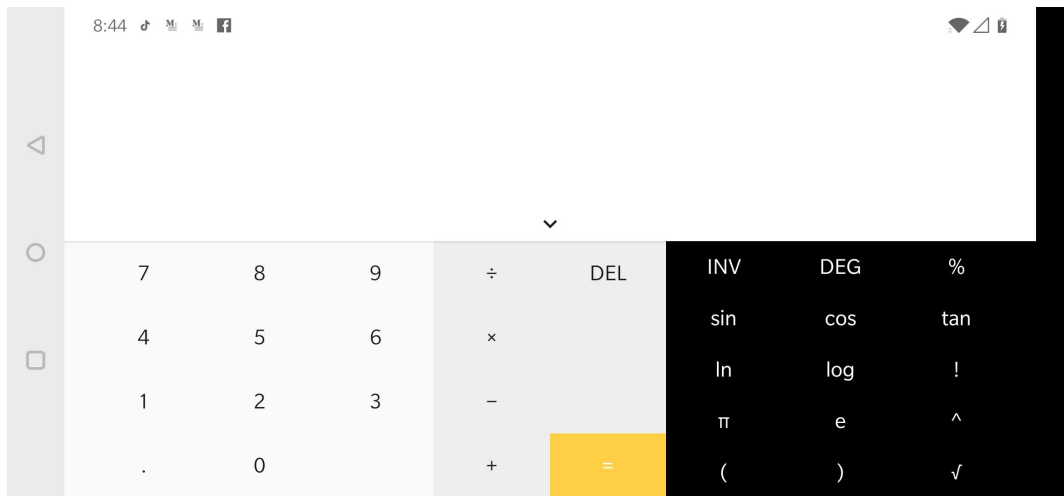
Why Fragment

- ❑ App Components/sub-Activity
- ❑ Reusability
- ❑ Adaptability
- ❑ Lifecycle

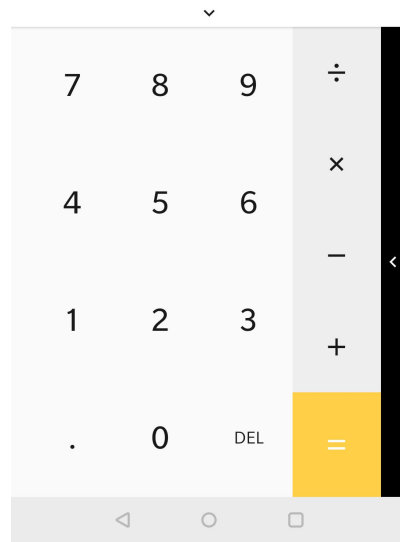
Fragment - Responsive Design



Fragment - Responsive Design



8:44



Defining a Fragment

□ Creating Layout file

```
1 <?xml version="1.0" encoding="utf-8"?>
2 <RelativeLayout
3     xmlns:android="http://schemas.android.com/apk/res/android"
4     android:layout_width="match_parent"
5     android:layout_height="match_parent">
6     <TextView
7         android:layout_width="wrap_content"
8         android:layout_height="wrap_content"
9         android:gravity="center"
10        android:text="FragmentA"
11        android:textSize="28sp"/>
12
13 </RelativeLayout>
```

□ Creating Java Class

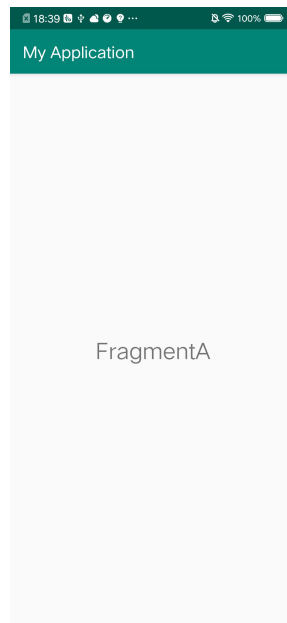
```
@Nullable
@Override
public View onCreateView(@NonNull LayoutInflater inflater, @Nullable ViewGroup container,
    @Nullable Bundle savedInstanceState) {
    return inflater.inflate(R.layout.frame_layout_a, container, attachToRoot: false);
}
```


Creating a Fragment

□ Embedding a Fragment in an Activity

```
activity_main.xml
1 <?xml version="1.0" encoding="utf-8"?>
2 <android.support.constraint.ConstraintLayout
3     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     tools:context=".MainActivity">
8     <fragment
9         android:id="@+id/FragmentA"
10        android:layout_width="match_parent"
11        android:layout_height="match_parent"
12        android:name="com.example.myapplication.FragmentA"/>
13
14
15 </android.support.constraint.ConstraintLayout>
```

```
public class MainActivity extends AppCompatActivity {
    @Override
    protected void onCreate(@Nullable Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
    }
}
```



Lifecycle

- onAttach/onDetach
- onCreate/onDestroy
- onCreateView/onDestroyView
- onActivityCreated
- onStart/onStop
- onResume/onPause



Lifecycle

□ Case 1: home

```
Vivo V1832A Android 9, API 28 | com.example.myapplication (22508) | Verbose | Q* FragmentA
2019-06-26 18:54:02.036 22508-22508/com.example.myapplication V/FragmentA: <-onAttach->
2019-06-26 18:54:02.036 22508-22508/com.example.myapplication V/FragmentA: <-onCreate->
2019-06-26 18:54:02.037 22508-22508/com.example.myapplication V/FragmentA: <-onCreateView->
2019-06-26 18:54:02.042 22508-22508/com.example.myapplication V/FragmentA: <-onViewCreated->
2019-06-26 18:54:02.047 22508-22508/com.example.myapplication V/FragmentA: <-onActivityCreated->
2019-06-26 18:54:02.047 22508-22508/com.example.myapplication V/FragmentA: <-onStart->
2019-06-26 18:54:02.050 22508-22508/com.example.myapplication V/FragmentA: <-onResume->
2019-06-26 18:54:08.648 22508-22508/com.example.myapplication V/FragmentA: <-onPause->
2019-06-26 18:54:08.995 22508-22508/com.example.myapplication V/FragmentA: <-onStop->
2019-06-26 18:54:11.618 22508-22508/com.example.myapplication V/FragmentA: <-onStart->
2019-06-26 18:54:11.625 22508-22508/com.example.myapplication V/FragmentA: <-onResume->
```

阶段1 启动应用

阶段2 按home键返回

阶段3 重新启动应用

□ Case 2: back

```
Vivo V1832A Android 9, API 28 | com.example.myapplication (22508) | Verbose | Q* FragmentA
2019-06-26 18:57:05.160 22508-22508/com.example.myapplication V/FragmentA: <-onAttach->
2019-06-26 18:57:05.160 22508-22508/com.example.myapplication V/FragmentA: <-onCreate->
2019-06-26 18:57:05.160 22508-22508/com.example.myapplication V/FragmentA: <-onCreateView->
2019-06-26 18:57:05.164 22508-22508/com.example.myapplication V/FragmentA: <-onViewCreated->
2019-06-26 18:57:05.168 22508-22508/com.example.myapplication V/FragmentA: <-onActivityCreated->
2019-06-26 18:57:05.169 22508-22508/com.example.myapplication V/FragmentA: <-onStart->
2019-06-26 18:57:05.171 22508-22508/com.example.myapplication V/FragmentA: <-onResume->
2019-06-26 18:57:07.163 22508-22508/com.example.myapplication V/FragmentA: <-onPause->
2019-06-26 18:57:07.523 22508-22508/com.example.myapplication V/FragmentA: <-onStop->
2019-06-26 18:57:07.528 22508-22508/com.example.myapplication V/FragmentA: <-onDestroyView->
2019-06-26 18:57:07.529 22508-22508/com.example.myapplication V/FragmentA: <-onDestroy->
2019-06-26 18:57:07.529 22508-22508/com.example.myapplication V/FragmentA: <-onDetach->
```

按back键返回主界面

Embedding a Fragment Dynamically

- FragmentManager
- FragmentTransaction

Public methods	
abstract void	<code>addOnBackStackChangeListener(FragmentManager.OnBackStackChangeListener listener)</code> Add a new listener for changes to the fragment back stack.
abstract <code>FragmentTransaction</code>	<code>beginTransaction()</code> Start a series of edit operations on the Fragments associated with this <code>FragmentManager</code> .
abstract void	<code>dump(String prefix, FileDescriptor fd, PrintWriter writer, String[] args)</code> Print the <code>FragmentManager</code> 's state into the given stream.
static void	<code>enableDebugLogging(boolean enabled)</code> Control whether the framework's internal fragment manager debugging logs are turned on.
abstract boolean	<code>executePendingTransactions()</code> After a <code>FragmentTransaction</code> is committed with <code>FragmentTransaction#commit</code> , it is scheduled to be executed asynchronously on the process's main thread.
abstract <code>Fragment</code>	<code>findFragmentById(int id)</code> Finds a fragment that was identified by the given id either when inflated from XML or as the container ID when added in a transaction.
abstract <code>Fragment</code>	<code>findFragmentByTag(String tag)</code> Finds a fragment that was identified by the given tag either when inflated from XML or as supplied when added in a transaction.
abstract <code>FragmentManager.BackStackEntry</code>	<code>getBackStackEntryAt(int index)</code> Return the <code>BackStackEntry</code> at index <code>index</code> in the back stack; where the item on the bottom of the stack has index 0.
abstract int	<code>getBackStackEntryCount()</code> Return the number of entries currently in the back stack.
abstract <code>Fragment</code>	<code>getFragment(Bundle bundle, String key)</code>

Public methods	
abstract <code>FragmentTransaction</code>	<code>add(Fragment fragment, String tag)</code> Calls <code>add(int, android.app.Fragment, java.lang.String)</code> with a 0 containerViewId.
abstract <code>FragmentTransaction</code>	<code>add(int containerViewId, Fragment fragment)</code> Calls <code>add(int, android.app.Fragment, java.lang.String)</code> with a null tag.
abstract <code>FragmentTransaction</code>	<code>add(int containerViewId, Fragment fragment, String tag)</code> Add a fragment to the activity state.
abstract <code>FragmentTransaction</code>	<code>addSharedElement(View sharedElement, String name)</code> Used with to map a View from a removed or hidden Fragment to a View from a shown or added Fragment.
abstract <code>FragmentTransaction</code>	<code>addToBackStack(String name)</code> Add this transaction to the back stack.
abstract <code>FragmentTransaction</code>	<code>attach(Fragment fragment)</code> Re-attach a fragment after it had previously been detached from the UI with <code>detach(android.app.Fragment)</code> .
abstract int	<code>commit()</code> Schedules a commit of this transaction.
abstract int	<code>commitAllowingStateLoss()</code> Like <code>commit()</code> but allows the commit to be executed after an activity's state is saved.
abstract void	<code>commitNow()</code> Commits this transaction asynchronously.
abstract void	<code>commitNowAllowingStateLoss()</code> Like <code>commitNow()</code> but allows the commit to be executed after an activity's state is saved.

需求：现在有fragmentA 和fragmentB, fragmentA 在初始化时就加入到Activity中，点击界面中的按钮替换为fragmentB

Embedding a Fragment Dynamically

□ Placeholder in Activity

```
<FrameLayout
    android:id="@+id/fragment_container"
    android:layout_width="match_parent"
    android:layout_height="match_parent">

</FrameLayout>
```

□ Add Fragments

```
//初始化时将fragment添加至activity中
FragmentManager manager = getSupportFragmentManager();
FragmentTransaction transaction = manager.beginTransaction();
transaction.add(R.id.fragment_container, new FragmentA());
transaction.commit();
```

Embedding a Fragment Dynamically

□ Replace Fragments

```
//点击按钮替换fragment
mButton.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        FragmentManager manager = getSupportFragmentManager();
        FragmentTransaction transaction = manager.beginTransaction();
        transaction.replace(R.id.fragment_container, new FragmentB());
        transaction.commit();
    }
});
```

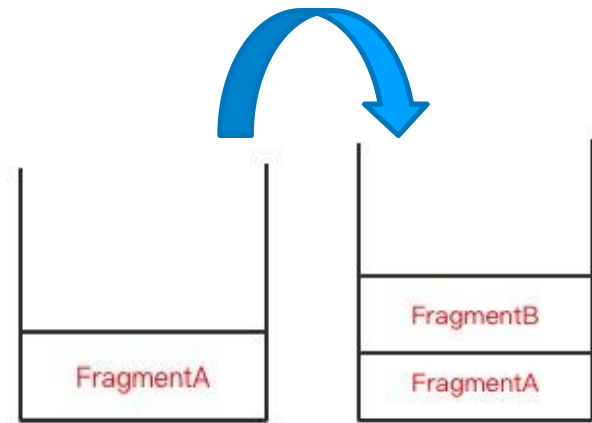
需求：切换至FragmentB 后可以回退到FragmentA，并且保存FragmentA中相关状态

Embedding a Fragment Dynamically

□ addToBackStack/popBackStack

```
30327-30327/com.example.myapplication V/FragmentB: <-----onAttach----->
30327-30327/com.example.myapplication V/FragmentB: <-----onCreate----->
30327-30327/com.example.myapplication V/FragmentA: <-----onPause----->
30327-30327/com.example.myapplication V/FragmentA: <-----onStop----->
30327-30327/com.example.myapplication V/FragmentA: <-----onDestroyView----->
30327-30327/com.example.myapplication V/FragmentB: <-----onCreateView----->
30327-30327/com.example.myapplication V/FragmentB: <-----onViewCreated----->
30327-30327/com.example.myapplication V/FragmentB: <-----onActivityCreated----->
30327-30327/com.example.myapplication V/FragmentB: <-----onStart----->
30327-30327/com.example.myapplication V/FragmentB: <-----onResume----->
```

```
30327-30327/com.example.myapplication V/FragmentB: <-----onPause----->
30327-30327/com.example.myapplication V/FragmentB: <-----onStop----->
30327-30327/com.example.myapplication V/FragmentB: <-----onDestroyView----->
30327-30327/com.example.myapplication V/FragmentB: <-----onDestroy----->
30327-30327/com.example.myapplication V/FragmentB: <-----onDetach----->
30327-30327/com.example.myapplication V/FragmentA: <-----onCreateView----->
30327-30327/com.example.myapplication V/FragmentA: <-----onViewCreated----->
30327-30327/com.example.myapplication V/FragmentA: <-----onActivityCreated----->
30327-30327/com.example.myapplication V/FragmentA: <-----onStart----->
30327-30327/com.example.myapplication V/FragmentA: <-----onResume----->
```



Communications - Arguments

```
public class StaticColorFragment extends Fragment {  
    private static final String KEY_COLOR = "key_color";  
  
    public static StaticColorFragment newInstance(int color) {  
        StaticColorFragment fragment = new StaticColorFragment();  
        Bundle args = new Bundle();  
        args.putInt(KEY_COLOR, color);  
        fragment.setArguments(args);  
        return fragment;  
    }  
  
    public View onCreateView(...) {  
        int color = Color.CYAN;  
        Bundle args = getArguments();  
        if (args != null) {  
            color = args.getInt(KEY_COLOR, Color.CYAN);  
        }  
        View view = new View(inflater.getContext());  
        view.setBackgroundColor(color);  
        return view;  
    }  
}
```

Fragment中创建静态newInstance方法，
使用setArguments()方法构造参数

从getArguments() 中解析bundle

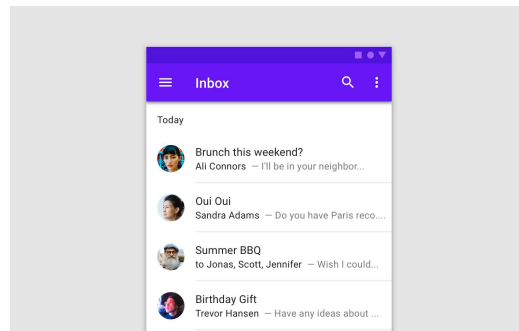
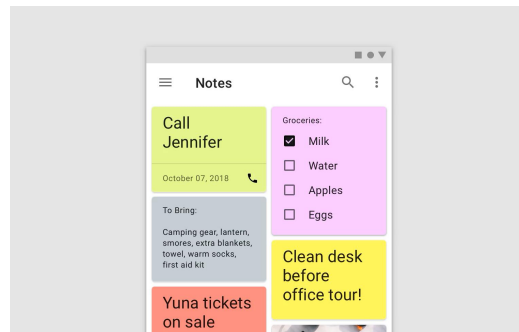
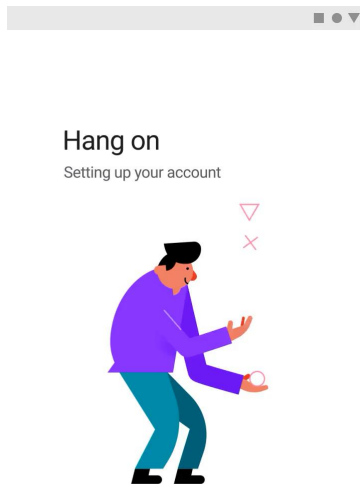
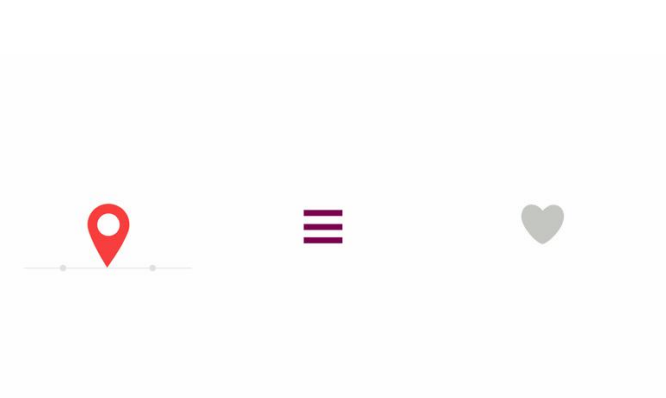


Demo- Master Detail

- Portrait
 - Master Activity: Item List
 - Detail Activity: Item Detail
- Landscape
 - One Activity: List & Detail

Animation

About Animation



Type of Animation

- ❑ FrameAnimation
- ❑ PropertyAnimation
- ❑ TweenAnimation



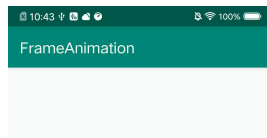
- View Animation: `android.view.animation`
- Property Animation: `android.animation`

FrameAnimation

```
<animation-list xmlns:android="http://schemas.android.com/apk/res/android" >
    <item
        android:drawable="@drawable/one"
        android:duration="500"/>
    <item
        android:drawable="@drawable/two"
        android:duration="500"/>
    <item
        android:drawable="@drawable/three"
        android:duration="500"/>
    <item
        android:drawable="@drawable/four"
        android:duration="500"/>
    <item
        android:drawable="@drawable/five"
        android:duration="500"/>
    <item
        android:drawable="@drawable/six"
        android:duration="500"/>
</animation-list>
```

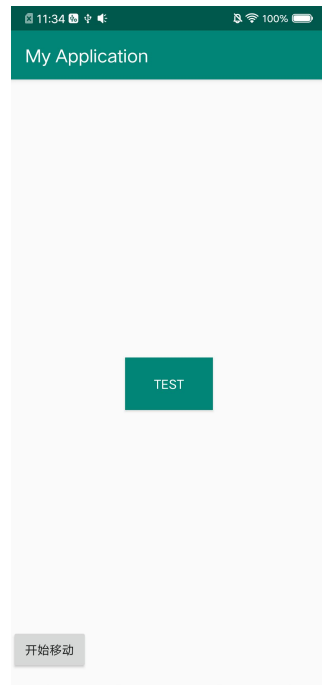
```
public class FrameAnimationActivity extends AppCompatActivity {
    private ImageView mImage;

    @Override
    protected void onCreate(@Nullable Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_frame_animation);
        setTitle("FrameAnimation");
        mImage = findViewById(R.id.image_frame);
        mImage.setBackgroundResource(R.drawable.anim_list);
        AnimationDrawable drawable = (AnimationDrawable) mImage.getBackground();
        drawable.start();
    }
}
```

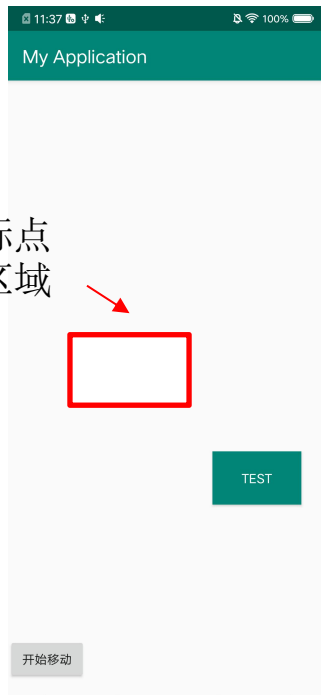


PropertyAnimation

- ❑ ObjectAnimator
- ❑ ValueAnimator
- ❑ AnimatorSet



实际点
击区域



PropertyAnimation

□ ObjectAnimator

代码示例:

```
<?xml version="1.0" encoding="utf-8"?>
<objectAnimator xmlns:android="http://schemas.android.com/apk/res/android"
    android:duration="8000"
    android:propertyName="rotation"
    android:interpolator="@android:anim/linear_interpolator"
    android:repeatCount="infinite"
    android:repeatMode="restart"
    android:valueFrom="0"
    android:valueTo="360" />
```

PropertyAnimation

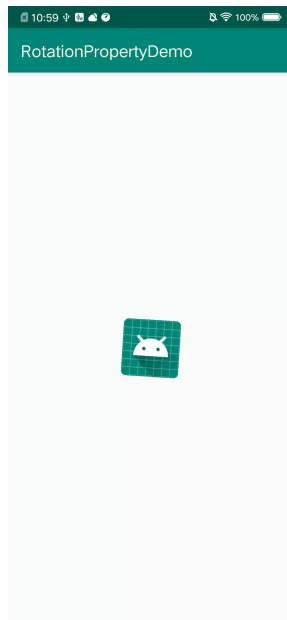
□ ObjectAnimator

代码示例:

```
ObjectAnimator animator = ObjectAnimator.ofFloat(findViewById(R.id.image_view),  
    propertyName: "rotation", ...values: 0, 360);  
animator.setRepeatCount(ValueAnimator.INFINITE);  
animator.setInterpolator(new LinearInterpolator());  
animator.setDuration(8000);  
animator.setRepeatMode(ValueAnimator.RESTART);  
animator.start();
```

其他属性:

- translationX/translationY
- rotationX/rotationY
- scaleX/scaleY
- alpha
-



PropertyAnimation

□ ObjectAnimator

```
ObjectAnimator objectAnimator = ObjectAnimator.ofFloat(mTestButton1, propertyName: "Height", ...values: 100);  
objectAnimator.start();  
ValueAnimator animator = ValueAnimator.ofFloat(0, 300);
```

The setter for this property does not match the expected signature (public void setHeight(float arg) more... (%F1)

ObjectAnimator 操作的属性必须具有**get/set**方法，不然无法生效



PropertyAnimation

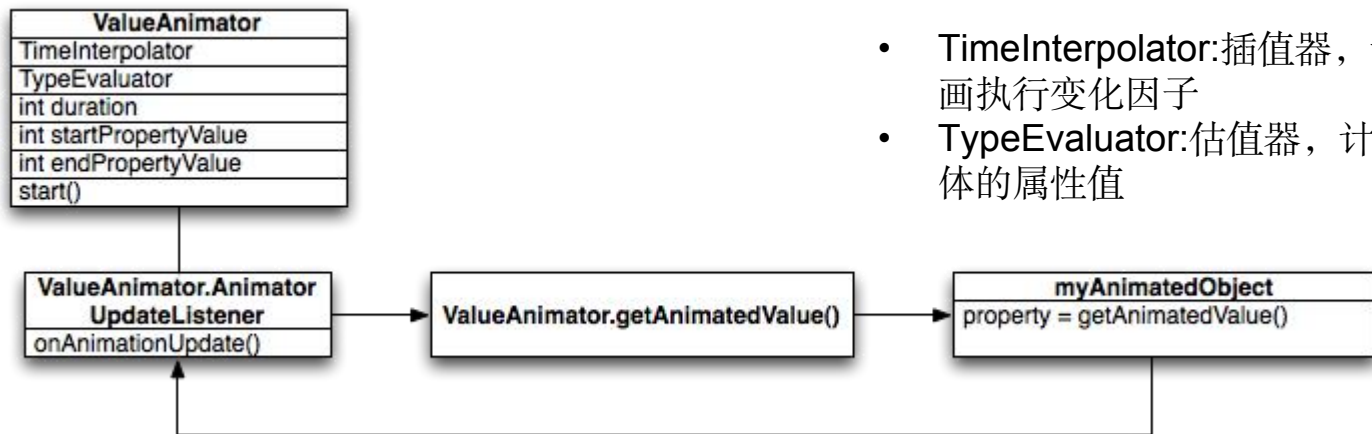
□ ObjectAnimator

使用包装类:

```
private static class WrapperView {  
    private View mTarget;  
  
    public WrapperView(View mTarget) {  
        this.mTarget = mTarget;  
    }  
    public int getHeight() {  
        return mTarget.getLayoutParams().height;  
    }  
    public void setHeight(int height) {  
        mTarget.getLayoutParams().height = height;  
        mTarget.requestLayout();  
    }  
}
```

PropertyAnimation

□ ValueAnimator



- **TimeInterpolator**:插值器，计算动画执行变化因子
- **TypeEvaluator**:估值器，计算具体的属性值

PropertyAnimation

□ ValueAnimator

```
ValueAnimator animation = ValueAnimator.ofFloat(0f, 100f);
animation.setDuration(1000);
animation.addUpdateListener(new ValueAnimator.AnimatorUpdateListener() {
    @Override
    public void onAnimationUpdate(ValueAnimator updatedAnimation) {
        float animatedValue = (float)updatedAnimation.getAnimatedValue();
        //todo use this value
        mView.setHeight(animatedValue);
    }
});
animation.start();
```

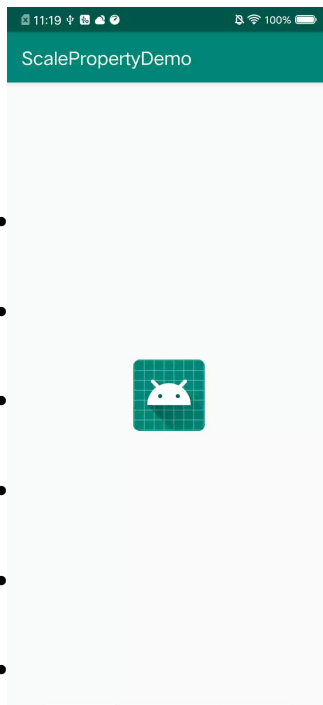
PropertyAnimation

□ AnimatorSet

1. 创建 breath.xml

```
<set xmlns:android="http://schemas.android.com/apk/res/android">
  <objectAnimator
    android:duration="1000"
    android:valueFrom="1.1"
    android:valueTo="0.9"
    android:propertyName="scaleX"
    android:interpolator="@android:anim/linear_interpolator"
    android:repeatMode="reverse"
    android:repeatCount="infinite" />

  <objectAnimator
    android:duration="1000"
    android:valueFrom="1.1"
    android:valueTo="0.9"
    android:propertyName="scaleY"
    android:interpolator="@android:anim/linear_interpolator"
    android:repeatMode="reverse"
    android:repeatCount="infinite" />
</set>
```



- alpha""scaleX"...
- infinite
- reverse / restart
-
-
-



PropertyAnimation

□ AnimatorSet

2.使用AnimationInflator加载

```
Animator animator = AnimatorInflater.loadAnimator(this, R.animator.breath);  
animator.setTarget(findViewById(R.id.image_view));  
animator.start();
```

PropertyAnimation

□ AnimatorSet

```
AnimatorSet setAnimation = new AnimatorSet();
```

//示例1

```
setAnimation.play(translateAnimation).after(alphaAnimation).before(rotateAnimation);  
setAnimation.play(rotateAnimation).before(scaleAnimation);
```

//示例2

```
setAnimation.playSequentially(alphaAnimation, translateAnimation, rotateAnimation,  
scaleAnimation);
```

//示例3

```
setAnimation.playTogether(alphaAnimation, translateAnimation, rotateAnimation,  
scaleAnimation);
```

TweenAnimation (不推荐使用)

- ❑ AlphaAnimation
- ❑ ScaleAnimation
- ❑ TranslateAnimation
- ❑ RotateAnimation
- ❑ AnimationSet

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

    <alpha
        android:duration="1000"
        android:fromAlpha="0.1"
        android:toAlpha="1.0" >
    </alpha>

</set>
```

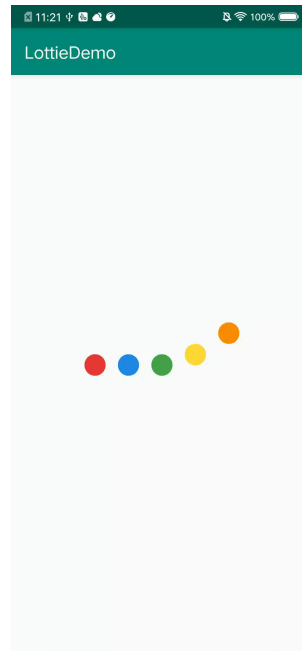

Lottie

□ build.gradle

```
dependencies {  
    implementation 'com.airbnb.android:lottie:$lottieVersion'  
}
```

□ LottieAnimationView

```
<com.airbnb.lottie.LottieAnimationView  
    android:id="@+id/animator_view"  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    app:lottie_loop="true"  
    app:lottie_autoPlay="true" />
```

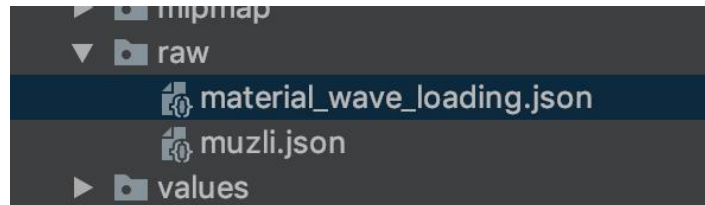


Demo-Lottie

```
<?xml version="1.0" encoding="utf-8"?>
<FrameLayout 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=".animation.LottieActivity">

    <com.airbnb.lottie.LottieAnimationView
        android:id="@+id/animation_view"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_gravity="center"
        app:lottie_autoPlay="true"
        app:lottie_loop="true"
        app:lottie_rawRes="@raw/material_wave_loading" />

</FrameLayout>
```





THANKS



ByteDance 字节跳动