

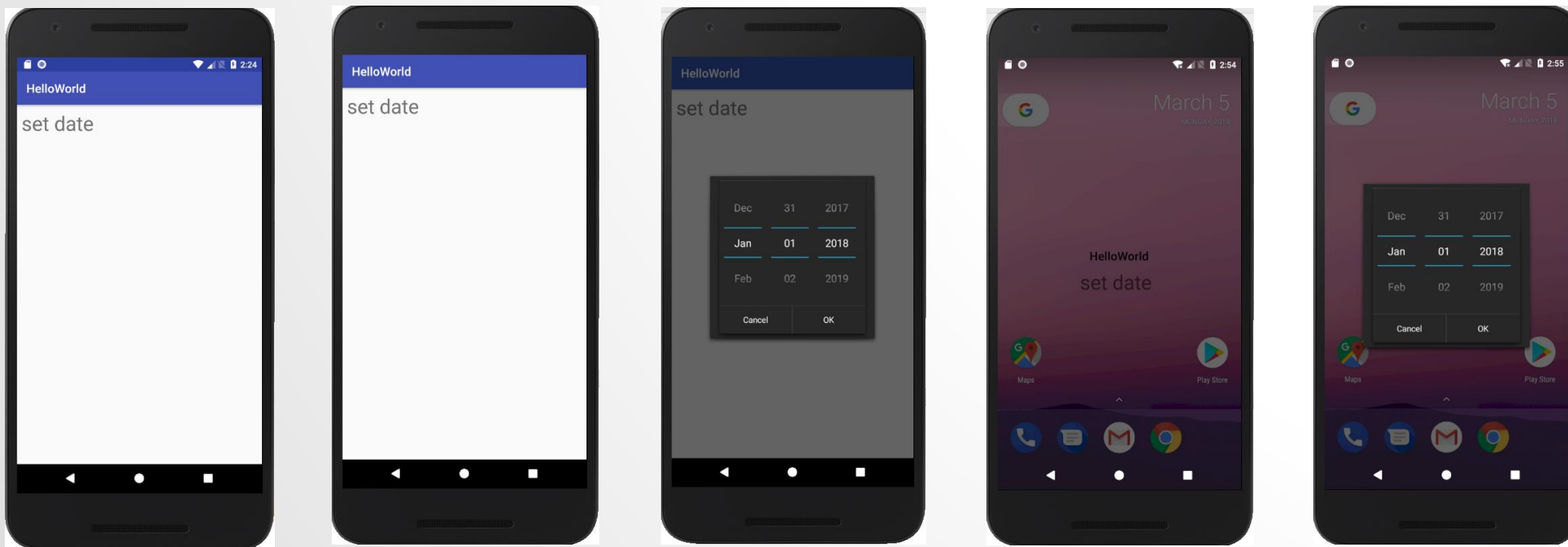
A faint, light gray world map is visible in the background, centered on the Atlantic Ocean. The map shows the outlines of continents and major landmasses.

ANDROID移动互联网应用开发

傅晓，河海大学计算机与信息学院
2019年3月

基本程序单元ACTIVITY

- Activity 是一个应用组件，用户可与其提供的屏幕进行交互，以执行拨打电话、拍摄照片、发送电子邮件或查看地图等操作。
- 每个 Activity 都会获得一个用于绘制其用户界面的窗口。窗口通常会充满屏幕，但也可小于屏幕并浮动在其他窗口之上。



基本程序单元ACTIVITY

- 一个应用通常由多个彼此松散联系的 Activity 组成。一般会指定应用中的某个 Activity 为“主”Activity，即首次启动应用时呈现给用户的那个 Activity。而且每个 Activity 均可启动另一个 Activity，以便执行不同的操作。每次新 Activity 启动时，前一 Activity 便会停止，但系统会在堆栈（“返回栈”）中保留该 Activity。
- 当新 Activity 启动时，系统会将其推送到返回栈上，并取得用户焦点。返回栈遵循基本的“后进先出”堆栈机制，因此，当用户完成当前 Activity 并按“返回”按钮时，系统会从堆栈中将其弹出（并销毁），然后恢复前一 Activity。

基本程序单元ACTIVITY

- 当一个 Activity 因某个新 Activity 启动而停止时，系统会通过该 Activity 的生命周期回调方法通知其这一状态变化。
- Activity 因状态变化——系统是创建 Activity、停止 Activity、恢复 Activity 还是销毁 Activity——而收到的回调方法可能有若干种，每一种回调都会为您提供执行与该状态变化相应的特定操作的机会。例如，停止时，您的 Activity 应释放任何大型对象，例如网络或数据库连接。当 Activity 恢复时，您可以重新获取所需资源，并恢复执行中断的操作。这些状态转变都是 Activity 生命周期的一部分。

基本程序单元ACTIVITY

```
public class Activity extends Application.Context {
    protected void onCreate(Bundle savedInstanceState);

    protected void onStart();

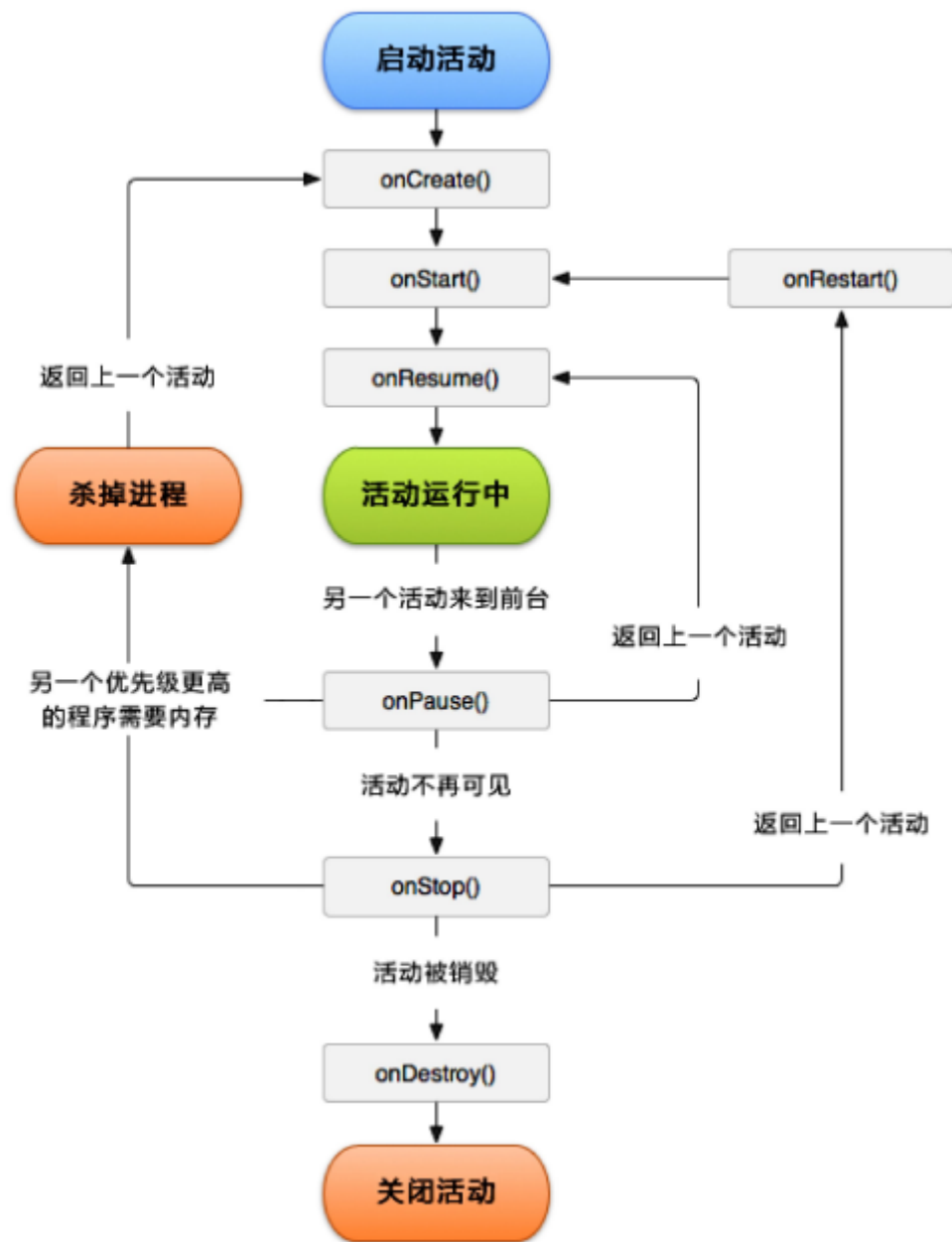
    protected void onRestart();

    protected void onResume();

    protected void onPause();

    protected void onStop();

    protected void onDestroy();
}
```



基本程序单元ACTIVITY

- Activity 的整个生命周期发生在 onCreate() 调用与 onDestroy() 调用之间。
- 您的 Activity 应在 onCreate() 中执行“全局”状态设置（例如定义布局），并释放 onDestroy() 中的所有其余资源。
- 例如，如果您的 Activity 有一个在后台运行的线程，用于从网络上下载数据，它可能会在 onCreate() 中创建该线程，然后在 onDestroy() 中停止该线程。

基本程序单元ACTIVITY

- Activity 的可见生命周期发生在 onStart() 调用与 onStop() 调用之间。
- 在这段时间，用户可以在屏幕上看到 Activity 并与其交互。例如，当一个新 Activity 启动，并且此 Activity 不再可见时，系统会调用 onStop()。您可以在调用这两个方法之间保留向用户显示 Activity 所需的资源。
- 例如，您可以在 onStart() 中注册一个 BroadcastReceiver 以监控影响 UI 的变化，并在用户无法再看到您显示的内容时在 onStop() 中将其取消注册。在 Activity 的整个生命周期，当 Activity 在对用户可见和隐藏两种状态中交替变化时，系统可能会多次调用 onStart() 和 onStop()。

基本程序单元ACTIVITY

- Activity 的前台生命周期发生在 onResume() 调用与 onPause() 调用之间。
- 在这段时间，Activity 位于屏幕上的所有其他 Activity 之前，并具有用户输入焦点。Activity 可频繁转入和转出前台 — 例如，当设备转入休眠状态或出现对话框时，系统会调用 onPause()。
- 由于此状态可能经常发生转变，因此这两个方法中应采用适度轻量级的代码，以避免因转变速度慢而让用户等待。

基本程序单元ACTIVITY

```
package com.example.bishop.helloworld;

import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.util.Log;

public class MainActivity extends AppCompatActivity {

    String msg = "Android: ";

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        Log.d(msg, msg: "MainActivity onCreate");
    }

    @Override
    protected void onStart() {
        super.onStart();
        Log.d(msg, msg: "MainActivity onStart");
    }

    @Override
    protected void onResume() {
        super.onResume();
        Log.d(msg, msg: "MainActivity onResume");
    }
```

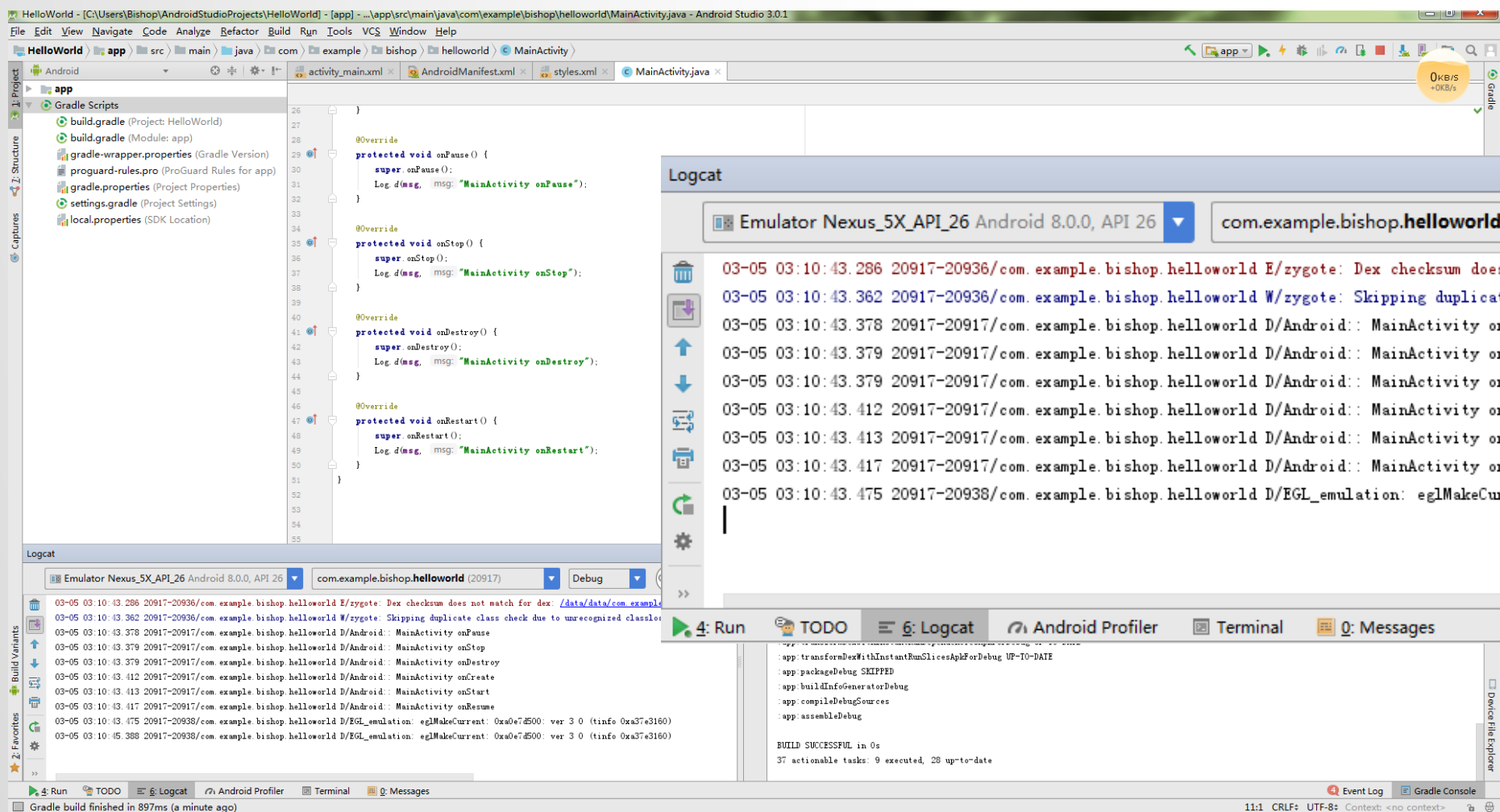
```
@Override
protected void onPause() {
    super.onPause();
    Log.d(msg, msg: "MainActivity onPause");
}

@Override
protected void onStop() {
    super.onStop();
    Log.d(msg, msg: "MainActivity onStop");
}

@Override
protected void onDestroy() {
    super.onDestroy();
    Log.d(msg, msg: "MainActivity onDestroy");
}

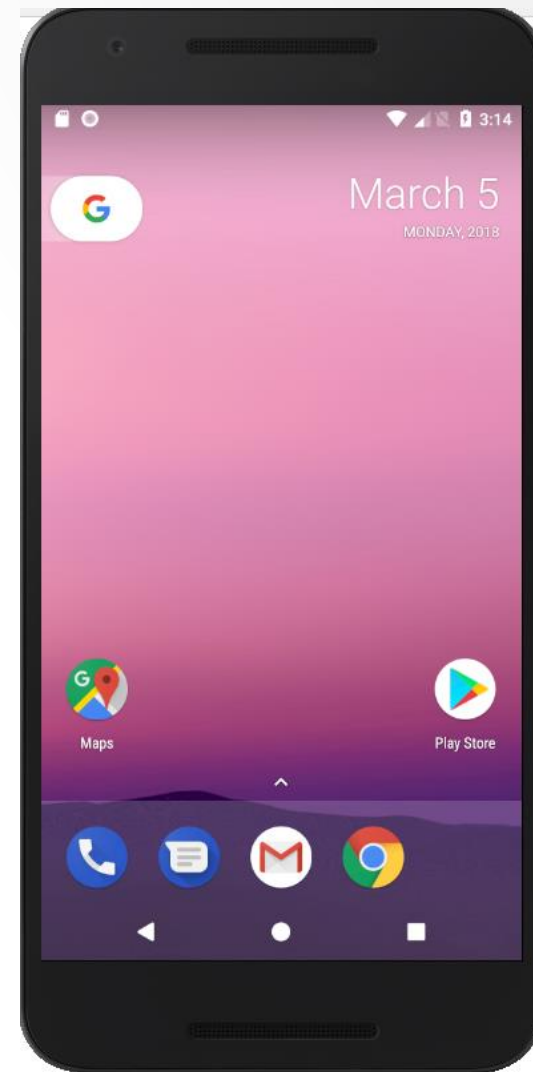
@Override
protected void onRestart() {
    super.onRestart();
    Log.d(msg, msg: "MainActivity onRestart");
}
```

基本程序单元ACTIVITY



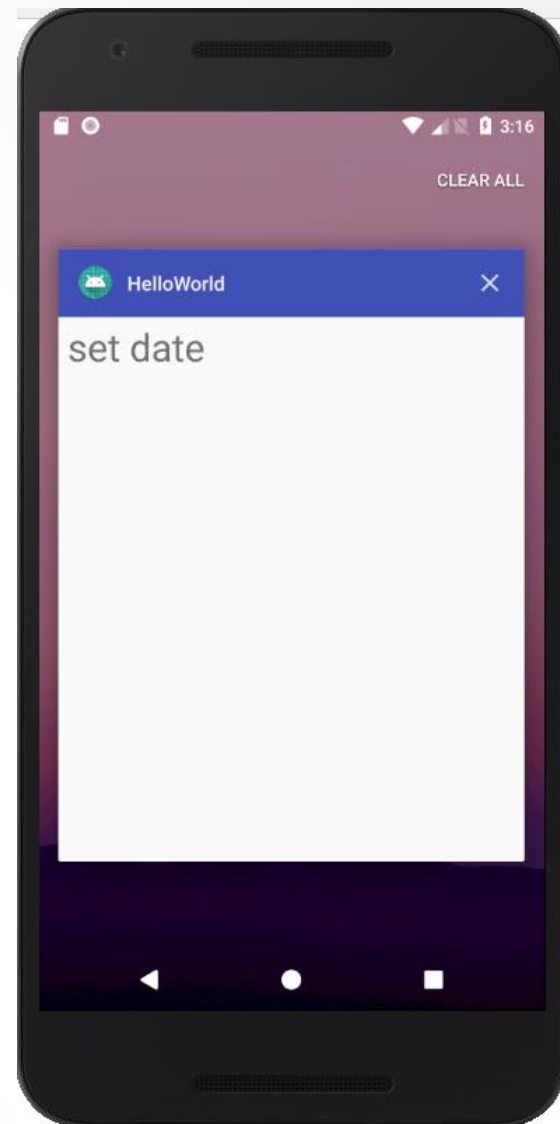
基本程序单元ACTIVITY

- 点击AVD的Home键:
- 03-05 03:12:22.156 20917-20917/com.example.bishop.helloworld D/Android:: MainActivity onPause
- 03-05 03:12:22.165 20917-20917/com.example.bishop.helloworld D/Android:: MainActivity onStop



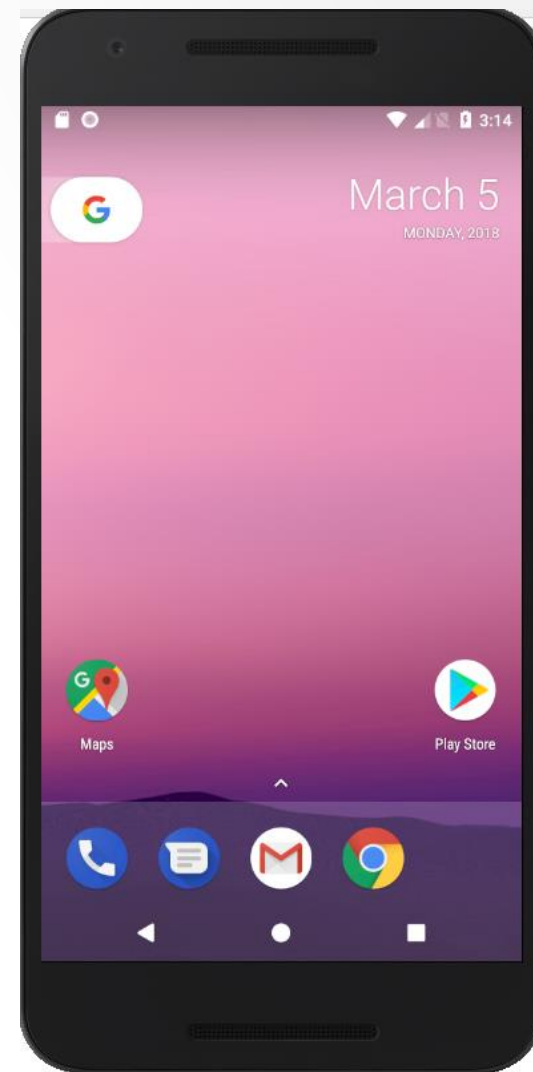
基本程序单元ACTIVITY

- 点击AVD的Menu键，选择刚才的Activity:
- 03-05 03:15:39.232 20917-20917/com.example.bishop.helloworld D/Android:: MainActivity onRestart
- 03-05 03:15:39.234 20917-20917/com.example.bishop.helloworld D/Android:: MainActivity onStart
- 03-05 03:15:39.235 20917-20917/com.example.bishop.helloworld D/Android:: MainActivity onResume



基本程序单元ACTIVITY

- 点击AVD的Menu键，选择Back键：
- 03-05 03:17:57.850 21597-21597/com.example.bishop.helloworld D/Android:: MainActivity onPause
- 03-05 03:17:58.271 21597-21597/com.example.bishop.helloworld D/Android:: MainActivity onStop
- 03-05 03:17:58.271 21597-21597/com.example.bishop.helloworld D/Android:: MainActivity onDestroy



基本程序单元ACTIVITY

- onCreate: Activity创建时调用。
- onStart: 可见未获取焦点, 无法与之交互。
- onResume: 可见已获取焦点, 可与之交互。
- onPause: 可见, 失去焦点。
- onStop: 不可见。
- onRestart: Activity重启。
- onDestroy: Activity被销毁。

基本程序单元ACTIVITY

-

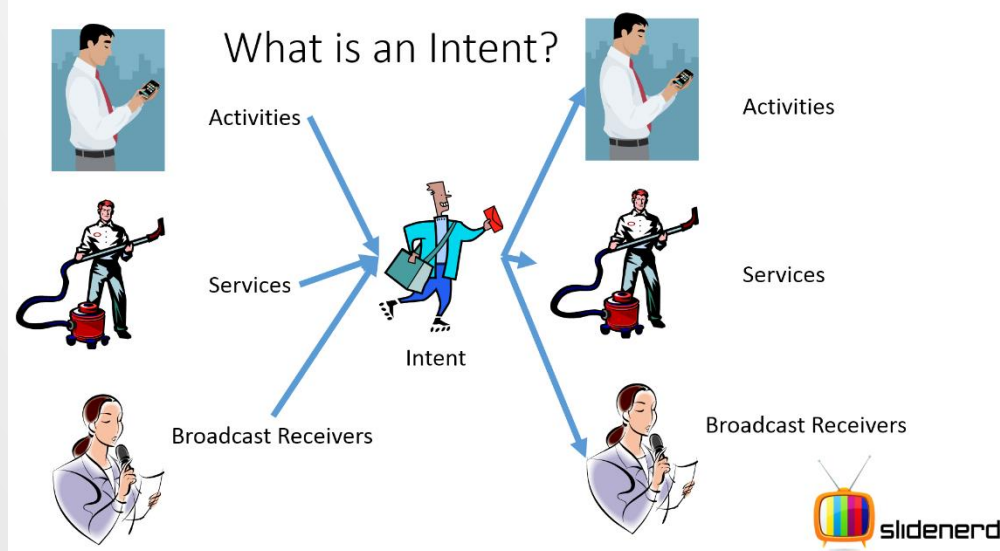
方法	说明	是否能 事后终 止？	后接
<u>onCreate()</u>	首次创建 Activity 时调用。您应该在此方法中执行所有正常的静态设置 — 创建视图、将数据绑定到列表等等。系统向此方法传递一个 Bundle 对象，其中包含 Activity 的上一状态，不过前提是捕获了该状态（请参阅后文的 保存 Activity 状态 ）。 始终后接 onStart()。	否	onStart()
<u>onRestart()</u>	在 Activity 已停止并即将再次启动前调用。 始终后接 onStart()	否	onStart()
<u>onStart()</u>	在 Activity 即将对用户可见之前调用。 如果 Activity 转入前台，则后接 onResume()，如果 Activity 转入隐藏状态，则后接 onStop()。	否	onResume() 或 onStop()

基本程序单元ACTIVITY

<u>onResume()</u>	在 Activity 即将开始与用户进行交互之前调用。此时，Activity 处于 Activity 堆栈的顶层，并具有用户输入焦点。 始终后接 onPause()。	否	onPause()
<u>onPause()</u>	当系统即将开始继续另一个 Activity 时调用。此方法通常用于确认对持久性数据的未保存更改、停止动画以及其他可能消耗 CPU 的内容，诸如此类。它应该非常迅速地执行所需操作，因为它返回后，下一个 Activity 才能继续执行。 如果 Activity 返回前台，则后接 onResume()，如果 Activity 转入对用户不可见状态，则后接 onStop()。	是	onResume() 或 onStop()
<u>onStop()</u>	在 Activity 对用户不再可见时调用。如果 Activity 被销毁，或另一个 Activity（一个现有 Activity 或新 Activity）继续执行并将其覆盖，就可能发生这种情况。 如果 Activity 恢复与用户的交互，则后接 onStart()，如果 Activity 被销毁，则后接 onDestroy()。	是	onRestart() 或 onDestroy()
<u>onDestroy()</u>	在 Activity 被销毁前调用。这是 Activity 将收到的最后调用。当 Activity 结束（有人对 Activity 调用了 <u>finish()</u> ），或系统为节省空间而暂时销毁该 Activity 实例时，可能会调用它。您可以通过 <u>isFinishing()</u> 方法区分这两种情形。	是	无

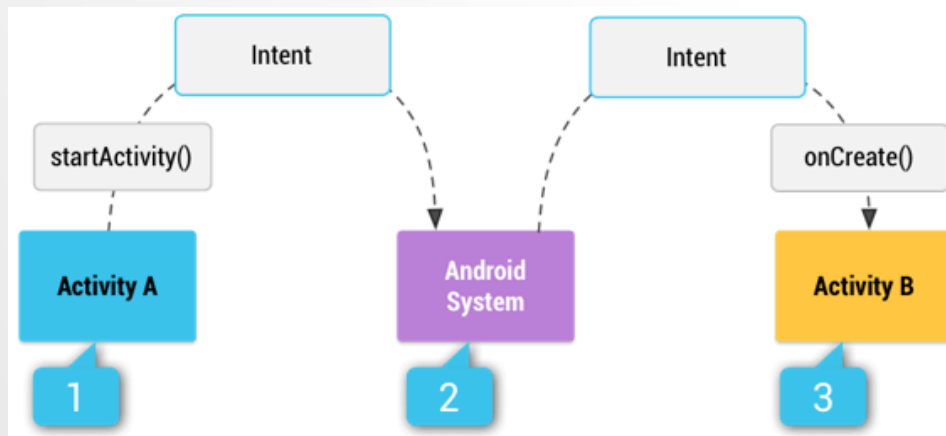
INTENT

- Intent的中文意思是“意图，意向”，在Android中提供了Intent机制来协助应用间的交互与通讯，Intent负责对应用中一次操作的动作、动作涉及数据、附加数据进行描述，Android则根据此Intent的描述，负责找到对应的组件，将Intent传递给调用的组件，并完成组件的调用。Intent不仅可用于应用程序之间，也可用于应用程序内部的Activity/Service之间的交互。因此，可以将Intent理解为不同组件之间通信的“媒介”专门提供组件互相调用的相关信息。



INTENT

- Intent 是一个将要执行的动作的抽象的描述，一般来说是作为参数来使用，由Intent来协助完成android各个组件之间的通讯。
- 比如说调用 startActivity()来启动一个activity,或者由 broadcastIntent()来传递给所有感兴趣的 BroadcastReceiver, 再或者由startService()/bindservice()来启动一个后台的service。
- 可以看出来，intent主要是用来启动其他的 activity 或者service，所以可以将intent理解成activity之间的粘合剂。



INTENT

- Intent可以启动一个Activity，也可以启动一个Service，还可以发起一个广播Broadcasts。

组件名称	方法名称
Activity	startActivity() startActivity()
Service	startService() bindService()
Broadcasts	sendBroadcasts() sendOrderedBroadcasts() sendStickyBroadcasts()

INTENT

- Intent的属性：
- component(组件)：目的组件。
- action (动作)：用来表现意图的行动。
- category (类别)：用来表现动作的类别。
- data (数据)：表示与动作要操纵的数据。
- type (数据类型)：对于data范例的描写。
- extras (扩展信息)：扩展信息。
- Flags (标志位)：期望这个意图的运行模式。

INTENT

- Component属性明确指定Intent的目标组件的类名称。
- ComponentName对象包含的构造器：
- ComponentName(String pkg,String cls)： 创建在pkg所在包下的cls类所对应的组件
- ComponentName(Context pkg,String cls)： 创建在pkg所在包下的cls类所对应的组件
- ComponentName(Context pkg,Class cls)创建在pkg所在包下的cls类所对应的组件
- 一个ComponentName需要指定包名和类名，这就可以唯一确定一个组件类。

INTENT

- 另外Intent可以根据给定的组件类去启动特定的组件，Intent有下面的方法：
- `setClass(Context packageContext, Class cls)`：设置该Intent将要启动的组件对应的类。
- `setClassName(Context packageContext, String className)`：设置该Intent将要启动的组件对应的类。
- `setClassName(String packageContext, String className)`：设置该Intent将要启动的组件对应的类。

INTENT

- Action 是一个用户定义的普通字符串，代表该Intent索要完成的一个抽象“动作”。用于描述一个 Android 应用程序组件，一个 Intent Filter 可以包含多个 Action。在 AndroidManifest.xml 的Activity 定义时，可以在其 节点指定一个 Action列表用于标识 Activity 所能接受的“动作”。
- Action要完成的只是一个抽象的动作，这个动作具体由哪个组件 (Activity,Service,BroadcastReceiver)来完成，Action这个字符不管。比如Android提供的标准Action: Intent.ACTION_VIEW，它只表示一个抽象的查看动作，但具体什么，启动哪个Activity来看，Intent.ACTION_VIEW并不知道---这取决于Activity的配置，只要某个Activity的配置中包含了该ACTION_VIEW，该Activity就有可能被启动。

INTENT

- 常见的Activity Action:

常量名称	常量值	意义
ACTION_MAIN	android.intent.action.MAIN	应用程序入口
ACTION_VIEW	android.intent.action.VIEW	显示数据给用户
ACTION_ATTACH_DATA	android.intent.action.ATTACH_DATA	指明附加信息给其他地方的一些数据
ACTION_EDIT	android.intent.action.EDIT	显示可编辑的数据
ACTION_PICK	android.intent.action.PICK	选择数据
ACTION_CHOOSER	android.intent.action.CHOOSER	显示一个Activity选择器
ACTION_GET_CONTENT	android.intent.action.GET_CONTENT	获得内容
ACTION_DIAL	android.intent.action.DIAL	显示打电话面板
ACITON_CALL	android.intent.action.CALL	直接打电话
ACTION_SEND	android.intent.action.SEND	直接发短信
ACTION_SENDTO	android.intent.action.SENDTO	选择发短信
ACTION_ANSWER	android.intent.action.ANSWER	应答电话
ACTION_INSERT	android.intent.action.INSERT	插入数据
ACTION_DELETE	android.intent.action.DELETE	删除数据
ACTION_RUN	android.intent.action.RUN	运行数据
ACTION_SYNC	android.intent.action.SYNC	同步数据
ACTION_PICK_ACTIVITY	android.intent.action.PICK_ACTIVITY	选择Activity
ACTION_SEARCH	android.intent.action.SEARCH	搜索
ACTION_WEB_SEARCH	android.intent.action.WEB_SEARCH	Web搜索

INTENT

- 常见的Broadcast Action:

BroadcastIntent Action字符串常量	描述
ACTION_TIME_TICK	系统时间每过一分钟发出的广播
ACTION_TIME_CHANGED	系统时间通过设置发生了变化
ACTION_TIMEZONE_CHANGED	时区改变
ACTION_BOOT_COMPLETED	系统启动完毕
ACTION_PACKAGE_ADDED	新的应用程序apk包安装完毕
ACTION_PACKAGE_CHANGED	现有应用程序apk包改变
ACTION_PACKAGE_REMOVED	现有应用程序apk包被删除
ACTION_UID_REMOVED	用户id被删除

INTENT

- Category是要执行动作的目标所具有的特质或行为归类，在Intent对象中可添加任意多个category。
- 常见的category：

Category字符串常量	描述
CATEGORY_BROWSABLE	目标Activity能通过在网页浏览器中点击链接而激活（比如，点击浏览器中的图片链接）
CATEGORY_GADGET	表示目标Activity可以被内嵌到其他Activity当中
CATEGORY_HOME	目标Activity是HOME Activity，即手机开机启动后显示的Activity，或按下HOME键后显示的Activity
CATEGORY_LAUNCHER	表示目标Activity是应用程序中最优先被执行的Activity
CATEGORY_PREFERENCE	表示目标Activity是一个偏爱设置的Activity

INTENT

- Data, 也就是执行动作要操作的数据。Android中采用指向数据的一个URI来表示, 如在联系人应用中, 一个指向某联系人的URI可能为: `content://contacts/1`。
- Type, 即数据类型, 显式指定Intent的数据类型 (MIME)。一般Intent的数据类型能够根据数据本身进行判定, 但是通过设置这个属性, 可以强制采用显式指定的类型而不再进行推导。

INTENT

- 在AndroidManifest.xml的<data/>标签中包含了以下几种子元素，他们定义了url的匹配规则：
- android:scheme：匹配url中的前缀，除了“http”、“https”、“tel”...之外，我们可以定义自己的前缀。
- android:host：匹配url中的主机名部分，如“google.com”，如果定义为“*”则表示任意主机名。
- android:port：匹配url中的端口。
- android:path：匹配url中的路径。

INTENT

- `<activity android:name=".TargetActivity" >`
- `<intent-filter>`
- `<action android:name="com.scott.intent.action.TARGET"/>`
- `<category android:name="android.intent.category.DEFAULT"/>`
- `<data android:scheme="scott"`
`android:host="com.scott.intent.data" android:port="7788"`
`android:path="/target"/>`
- `</intent-filter>`
- `</activity>`

INTENT

- Intent的Action和Data组合例子：

Action属性	Data属性	说明
ACTION_VIEW	content://contacts/people/1	显示id为1的联系人信息
ACTION_DIAL	content://contacts/people/1	将id为1的联系人电话号码显示在拨号界面中
ACITON_VIEW	tel:123	显示电话为123的联系人信息
ACTION_VIEW	http://www.google.com	在浏览器中浏览该网站
ACTION_VIEW	file://sdcard/mymusic.mp3	播放MP3
ACTION_VIEW	geo:39.2456,116.3523	显示地图

INTENT

- Extras属性主要用于传递目标组件所需要的额外的数据。通过putExtras()方法设置。常用值如下所示：
- EXTRA_BCC：存放邮件密送人地址的字符串数组。
- EXTRA_CC：存放邮件抄送人地址的字符串数组。
- EXTRA_EMAIL：存放邮件地址的字符串数组。
- EXTRA_SUBJECT：存放邮件主题字符串。
- EXTRA_TEXT：存放邮件内容。
- EXTRA_KEY_EVENT：以KeyEvent对象方式存放触发Intent的按键。
- EXTRA_PHONE_NUMBER：存放调用ACTION_CALL时的电话号码。

INTENT

- Flag (标记) 用于为Intent添加额外控制标记。常用值如下所示:
- FLAG_ACTIVITY_BROUGHT_TO_FRONT: 这个标志一般不是由程序代码设置的, 如在launchMode中设置singleTask模式时系统帮你设定。
- FLAG_ACTIVITY_CLEAR_TOP: 如果目标Activity已经运行于当前的Task中, 则关闭Activity Stack中在此Activity上方的所有Activity, 然后将此Intent传递到该Activity实例内。
- FLAG_ACTIVITY_NEW_TASK: 将使Activity成为一个新Task的开始。
- FLAG_ACTIVITY_NO_ANIMATION: 这个标志将阻止系统进入下一个Activity时应用Activity迁移动画。
- FLAG_ACTIVITY_NO_HISTORY: 新的Activity将不再历史stack中保留。一旦离开, 此Activity就关闭了。

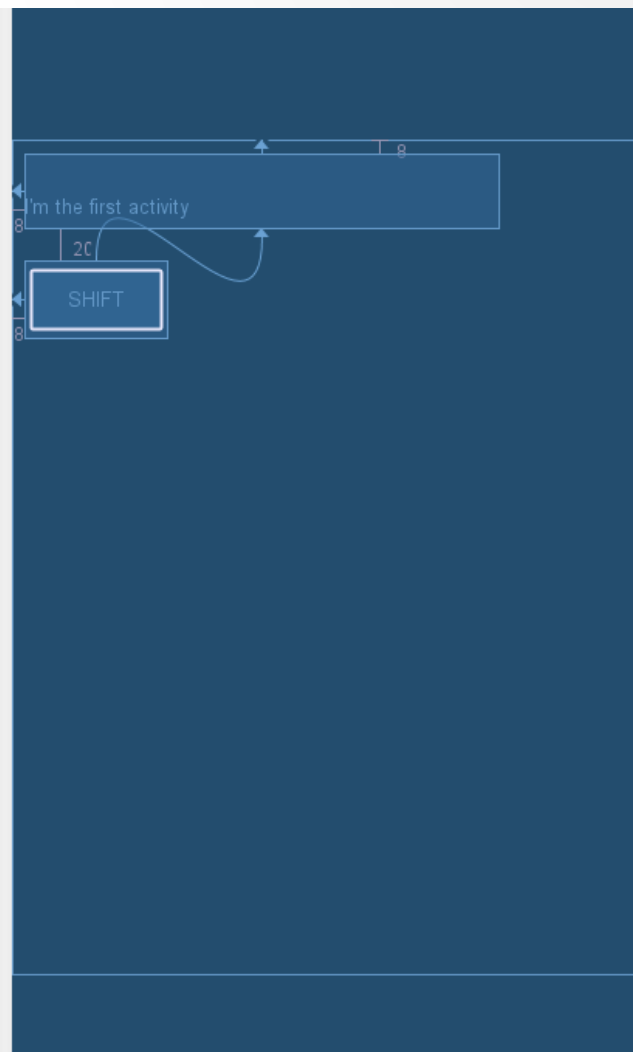
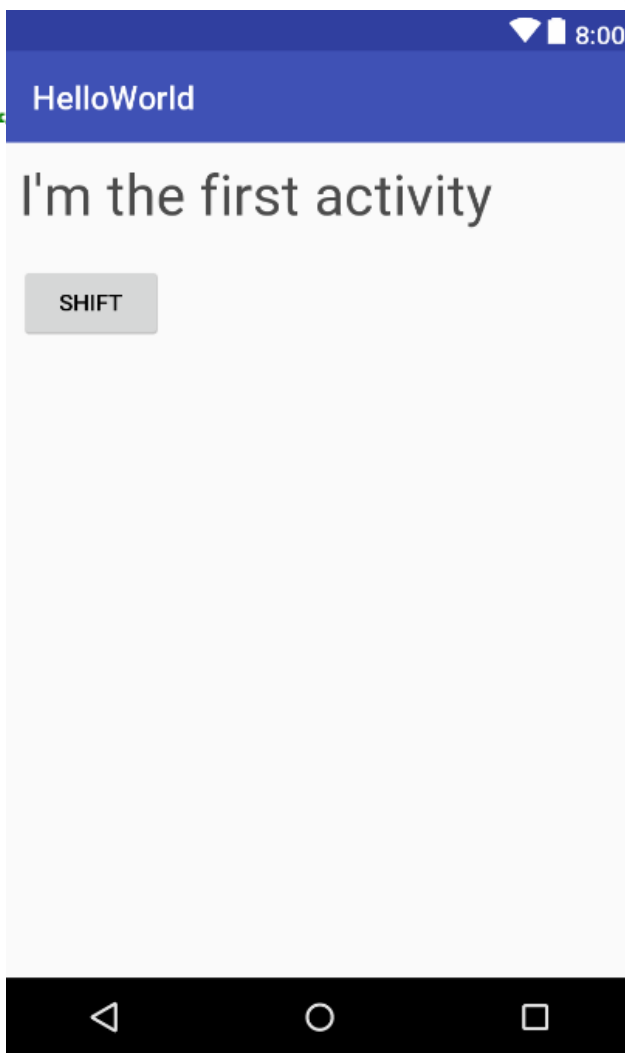
显式INTENT

```
<?xml version="1.0" encoding="utf-8"?>
<android.support.constraint.ConstraintLayout xmlns:android="http://schemas.android.com/apk
    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"
    android:orientation="vertical"
    tools:layout_editor_absoluteY="81dp">

    <TextView
        android:id="@+id/Date"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignParentTop="true"
        android:layout_centerHorizontal="true"
        android:layout_marginStart="8dp"
        android:layout_marginTop="8dp"
        android:text="I'm the first activity"
        android:textAppearance="@style/TextAppearance.AppCompat.Display1"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent" />

    <Button
        android:id="@+id/button"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginStart="8dp"
        android:layout_marginTop="20dp"
        android:text="shift"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/Date" />

</android.support.constraint.ConstraintLayout>
```



显式INTENT

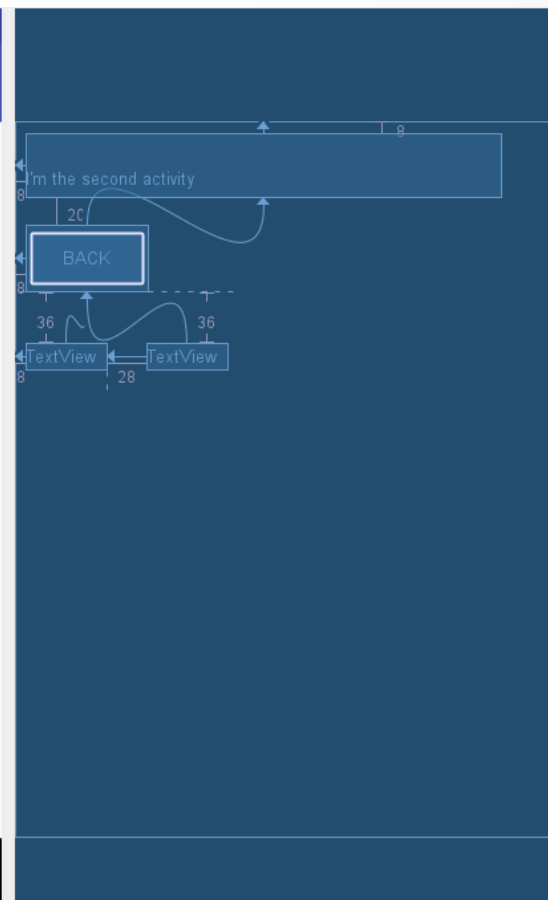
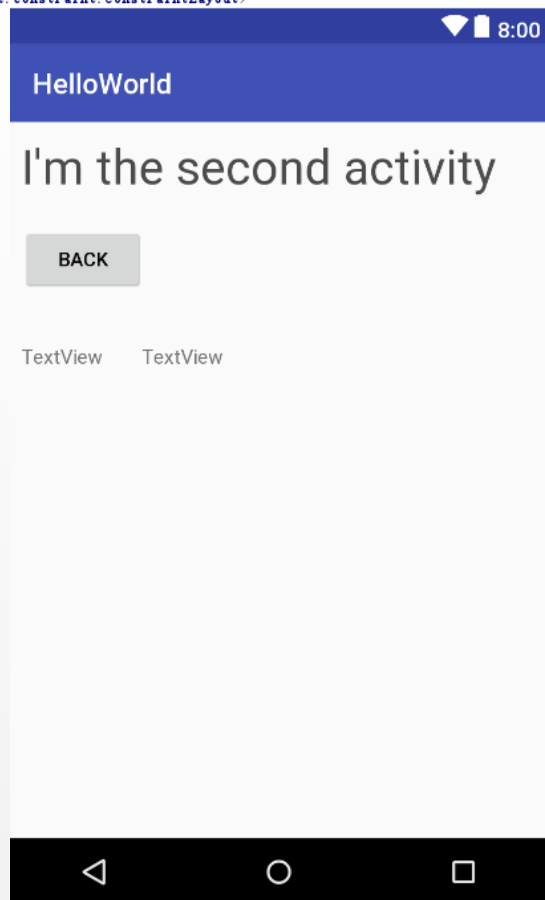
```
<?xml version="1.0" encoding="utf-8"?>
<android.support.constraint.ConstraintLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context="com.example.bishop.helloworld.Main2Activity">
    <TextView
        android:id="@+id/Date"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignParentTop="true"
        android:layout_centerHorizontal="true"
        android:layout_marginStart="8dp"
        android:layout_marginTop="8dp"
        android:text="I'm the second activity"
        android:textAppearance="@style/TextAppearance.AppCompat.Display1"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent" />

    <Button
        android:id="@+id/button"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginStart="8dp"
        android:layout_marginTop="20dp"
        android:text="back"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/Date" />

    <TextView
        android:id="@+id/textView1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginStart="8dp"
        android:layout_marginTop="36dp"
        android:text="TextView"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/button" />

    <TextView
        android:id="@+id/textView2"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
```

```
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginStart="28dp"
        android:layout_marginTop="36dp"
        android:text="TextView"
        app:layout_constraintStart_toEndOf="@+id/textView1"
        app:layout_constraintTop_toBottomOf="@+id/button" />
    </android.support.constraint.ConstraintLayout>
```



显式INTENT

```
package com.example.bishop.helloworld;
import android.content.Intent;
import android.support.v7.app.AppCompatActivity;
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);
        Button b = (Button)findViewById(R.id.button);
        b.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                Intent i = new Intent( packageContext: MainActivity.this, Main2Activity.class);
                i.putExtra( name: "name1", value: "Alice");
                i.putExtra( name: "name2", value: "Bob");
                startActivity(i);
            }
        });
    }
}
```

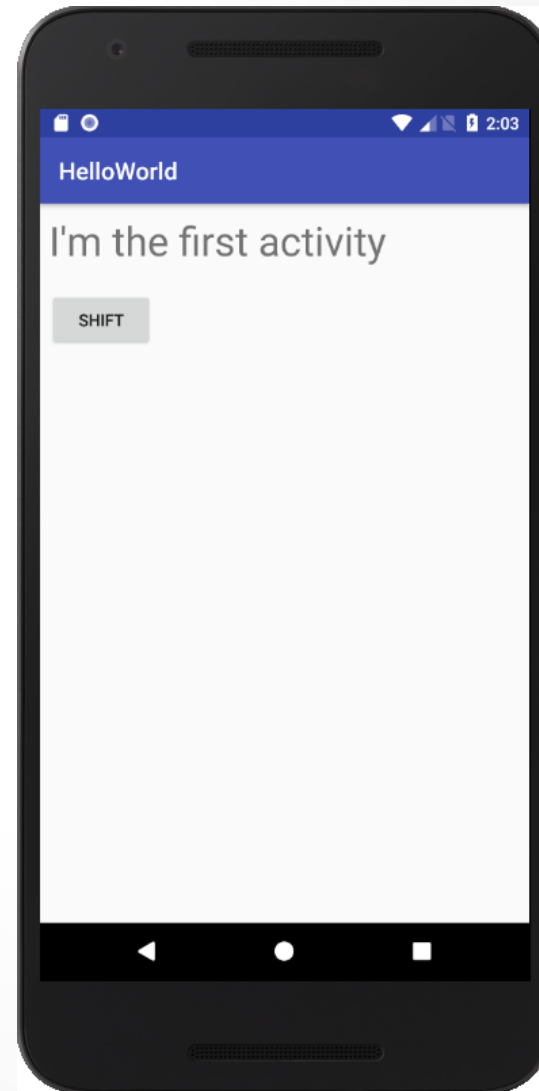
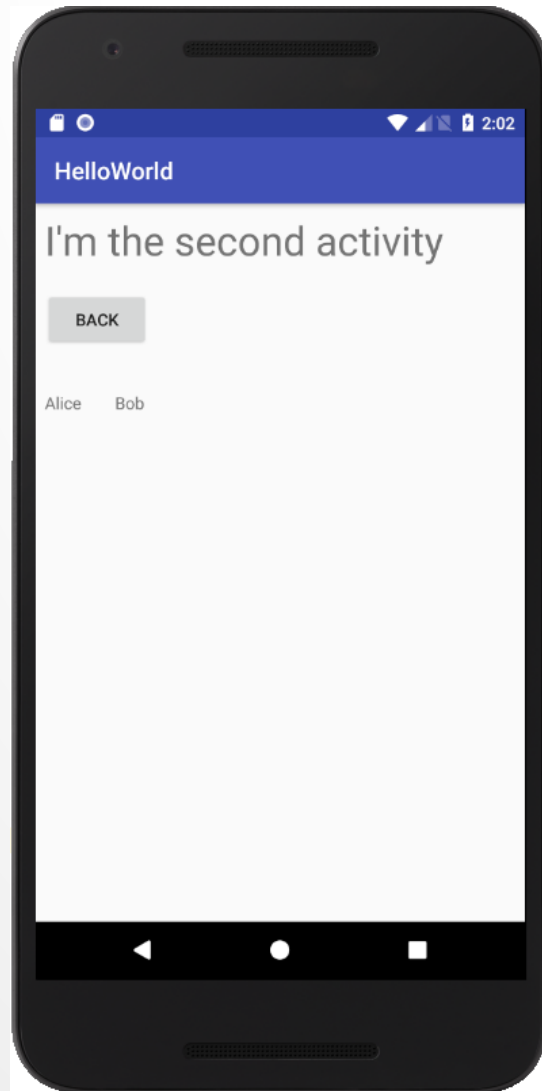
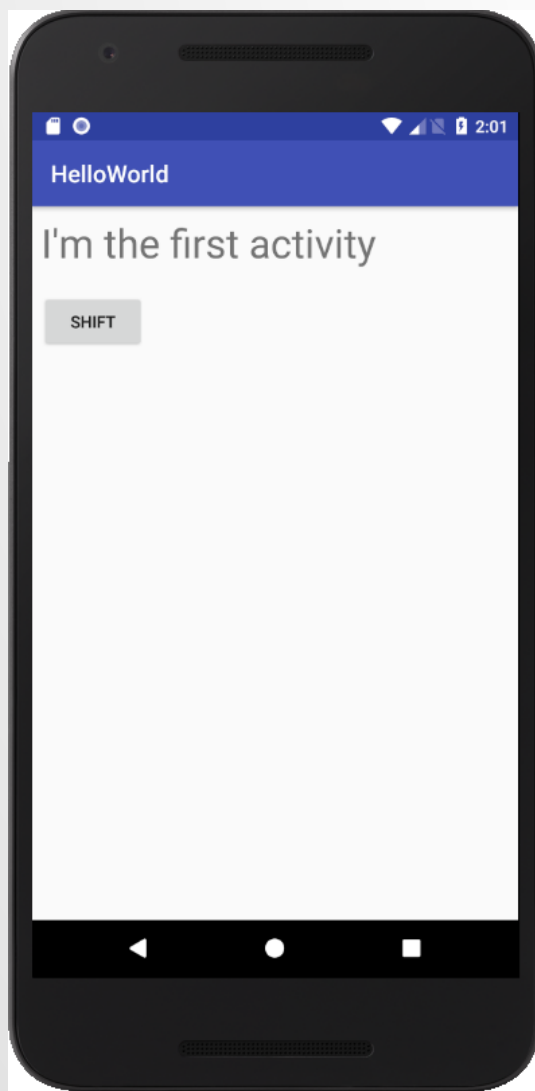
```
package com.example.bishop.helloworld;
import android.content.Intent;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.TextView;
import org.w3c.dom.Text;

public class Main2Activity extends AppCompatActivity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main2);

        Bundle bundle=getIntent().getExtras();
        String name1=bundle.getString( key: "name1");
        String name2=bundle.getString( key: "name2");
        TextView t1=(TextView)findViewById(R.id.textView1);
        TextView t2=(TextView)findViewById(R.id.textView2);
        t1.setText(name1);
        t2.setText(name2);
        Button b = (Button)findViewById(R.id.button);
        b.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                Intent i = new Intent( packageContext: Main2Activity.this, MainActivity.class);
                startActivity(i);
            }
        });
    }
}
```

显式INTENT



隐式INTENT

```
<?xml version="1.0" encoding="utf-8"?>
<android.support.constraint.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"
    android:orientation="vertical"
    tools:layout_editor_absoluteY="81dp">
```

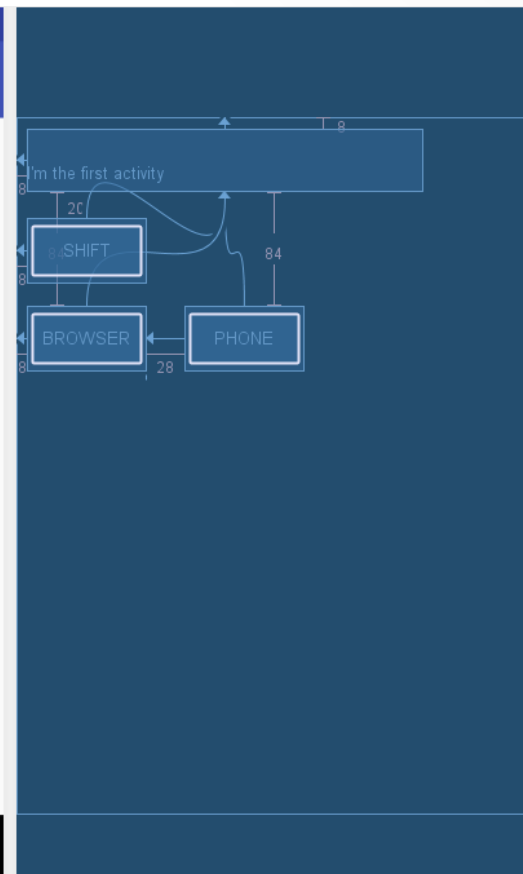
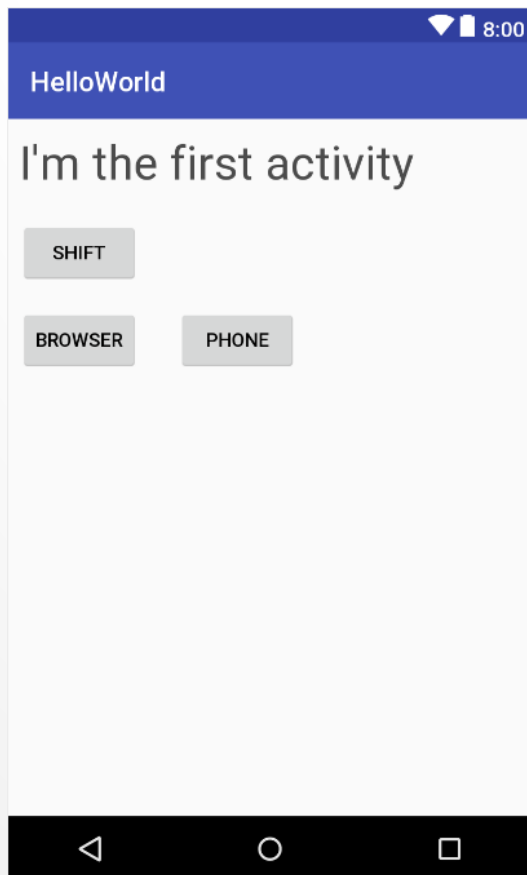
```
<TextView
    android:id="@+id/Date"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_alignParentTop="true"
    android:layout_centerHorizontal="true"
    android:layout_marginStart="8dp"
    android:layout_marginTop="8dp"
    android:text="I'm the first activity"
    android:textAppearance="@style/TextAppearance.AppCompat.Display1"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent" />
```

```
<Button
    android:id="@+id/button3"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginStart="8dp"
    android:layout_marginTop="84dp"
    android:text="Browser"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/Date" />
```

```
<Button
    android:id="@+id/button2"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginStart="28dp"
    android:layout_marginTop="84dp"
    android:text="phone"
    app:layout_constraintStart_toEndOf="@+id/button3"
    app:layout_constraintTop_toBottomOf="@+id/Date" />
```

```
<Button
    android:id="@+id/button"
```

```
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginStart="8dp"
    android:layout_marginTop="20dp"
    android:text="shift"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/Date" />
</android.support.constraint.ConstraintLayout>
```



隐式INTENT

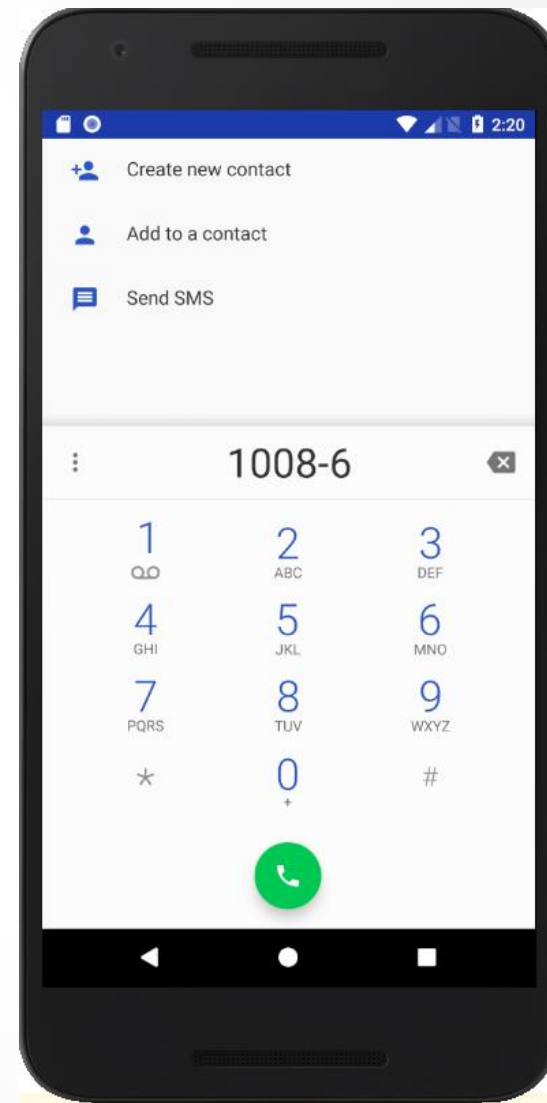
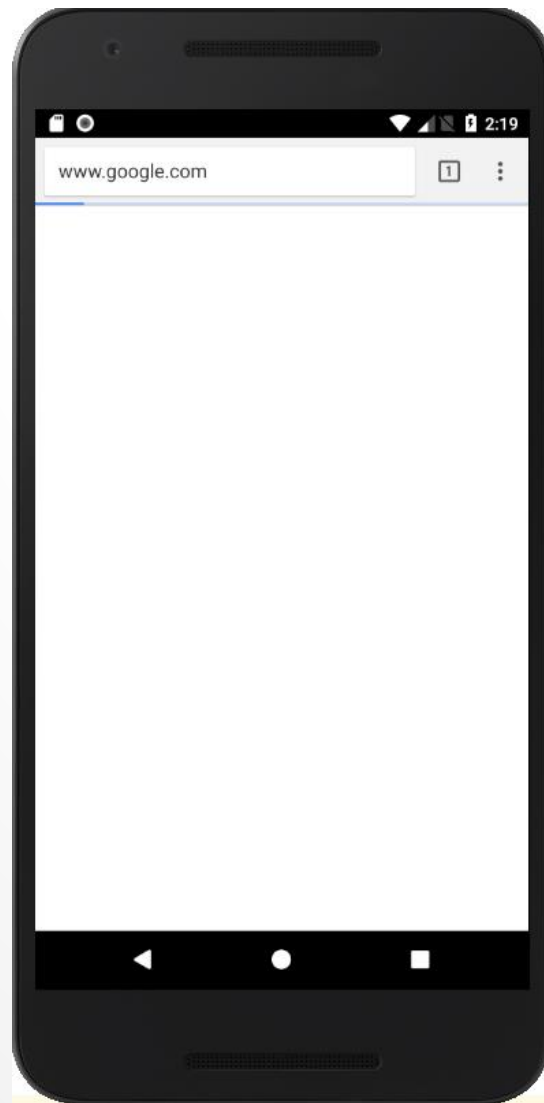
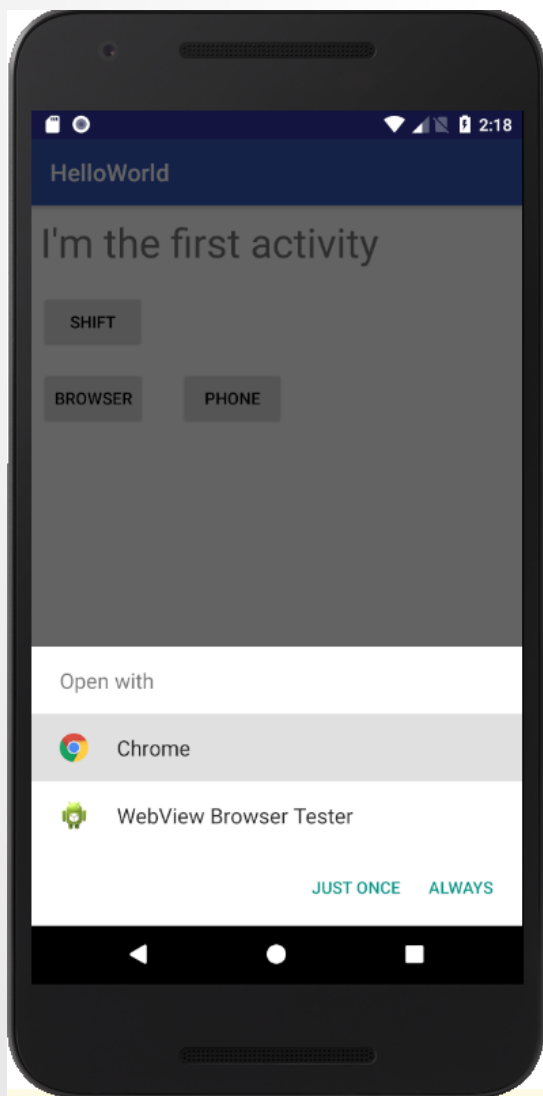
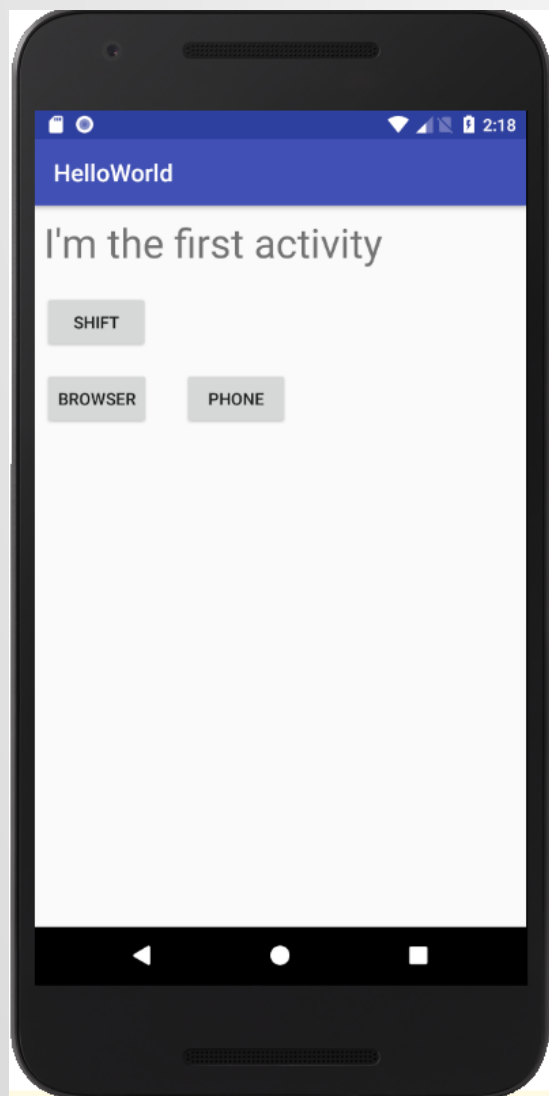
```
package com.example.bishop.helloworld;

import android.content.Intent;
import android.net.Uri;
import android.support.v7.app.AppCompatActivity;
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);
        Button b = (Button)findViewById(R.id.button);
        Button b2 = (Button)findViewById(R.id.button2);
        Button b3 = (Button)findViewById(R.id.button3);
        b.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                Intent i = new Intent(packageContext: MainActivity.this, Main2Activity.class);
                i.putExtra(name: "name1", value: "Alice");
                i.putExtra(name: "name2", value: "Bob");
                startActivity(i);
            }
        });
        b2.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                Intent i = new Intent(Intent.ACTION_VIEW, Uri.parse("tel:10086"));
                startActivity(i);
            }
        });
        b3.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                Intent i = new Intent(Intent.ACTION_VIEW, Uri.parse("http://www.google.com"));
                startActivity(i);
            }
        });
    }
}
```

隱式INTENT



INTENT FILTER

- IntentFilter翻译成中文就是“意图过滤器”，主要用来过滤隐式意图。当用户进行一项操作的时候，Android系统会根据配置的“意图过滤器”来寻找可以响应该操作的组件，服务。
- 例如当用户点击PDF文件的时候，Android系统就会通过设定好的意图过滤器，进行匹配测试。找到能够打开PDF文件的APP程序。

```
<activity android:name="com.example.testmain.ShowActivity" >
    <intent-filter>
        <action android:name="test.update.mydata" />
        <category android:name="my.test.show" />
        <data android:pathPattern=".*\\.jpg" android:scheme="http" />
    </intent-filter>
</activity>
```

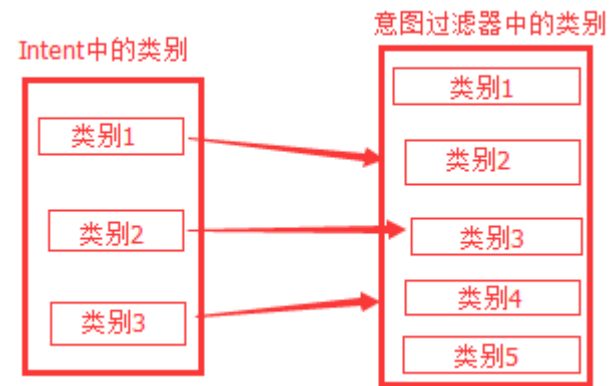

INTENT FILTER

- Android系统会根据我们配置的Intent Filter（意图过滤器），来进行匹配测试。匹配的时候，只会考虑三个方面：动作、数据（URI以及数据类型）和类别。
- 也就是说Android系统会进行“动作测试”，“数据测试”，“类别测试”，来寻找可以响应隐式意图的组件或服务。
- 另外，当对其他App程序开放组件和服务的时候也需要配置Intent Filter（意图过滤器），一个Activity可以配置多个<intent-filter>。

INTENT FILTER

- 对应<intent-filter>中的<action/>标签;
- 1, 如果<intent-filter>标签中有多个<action/>, 那么Intent请求的Action, 只要匹配其中的一条<action/>就可以通过了这条<intent-filter>的动作测试。
- 2, 如果<intent-filter>中没有包含任何<action/>, 那么无论什么Intent请求都无法和这条<intent-filter>匹配。
- 3, 如果Intent请求中没有设定Action(动作), 那么这个Intent请求就将顺利地通过<intent-filter>的动作测试 (前提是<intent-filter>中必须包含有<action/>, 否则与第二条冲突) 。

INTENT FILTER



- 对应<intent-filter>中的<category />标签;
- Intent中的类别必须全部匹配<intent-filter>中的<category />, 但是<intent-filter>中多余的<category />将不会导致匹配失败。
- 例如: Intent中有3个类别, 而意图过滤器中定义了5个, 如果Intent中的3个类别都与过滤器中的匹配, 那么过滤器中的另外2个, 将不会导致类别测试失败。
- 注意有一个例外, Android把所有传给startActivity()的隐式意图当作他们包含至少一个类别: "android.intent.category.DEFAULT" (CATEGORY_DEFAULT常量)。因此, 想要接收隐式意图的活动必须在它们的意图过滤器中包含"android.intent.category.DEFAULT"。

INTENT FILTER

- 对应<intent-filter>中的<data>标签;
- <data>元素指定了可以接受的Intent传过来的数据URI和数据类型, 当一个意图对象中的URI被用来和一个过滤器中的URI比较时, 比较的是URI的各个组成部分。
- 例如:
- 如果过滤器仅指定了一个scheme, 所有该scheme的URIs都能够和这个过滤器相匹配;
- 如果过滤器指定了一个scheme、主机名但没有路经部分, 所有具有相同scheme和主机名的URIs都可以和这个过滤器相匹配, 而不管它们的路经;
- 如果过滤器指定了一个scheme、主机名和路经, 只有具有相同scheme、主机名和路经的URIs才可以和这个过滤器相匹配。
- 当然, 一个过滤器中的路径规格可以包含通配符, 这样只需要部分匹配即可。

INTENT FILTER

```
<?xml version="1.0" encoding="utf-8"?>
<android.support.constraint.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"
    android:orientation="vertical"
    tools:layout_editor_absoluteT="81dp">
```

```
<TextView
    android:id="@+id/Date"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_alignParentTop="true"
    android:layout_centerHorizontal="true"
    android:layout_marginStart="8dp"
    android:layout_marginTop="8dp"
    android:text="I'm the first activity"
    android:textAppearance="@style/TextAppearance.AppCompat.Display1"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent" />
```

```
<Button
    android:id="@+id/button3"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginStart="8dp"
    android:layout_marginTop="16dp"
    android:text="Start broeser with launch action"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/button2" />
```

```
<Button
    android:id="@+id/button2"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginStart="8dp"
    android:layout_marginTop="16dp"
    android:text="Start broeser with view action"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/button" />
```

```
<Button
    android:id="@+id/button"
```

```
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginStart="8dp"
    android:layout_marginTop="20dp"
    android:text="shift"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/Date" />
</android.support.constraint.ConstraintLayout>
```

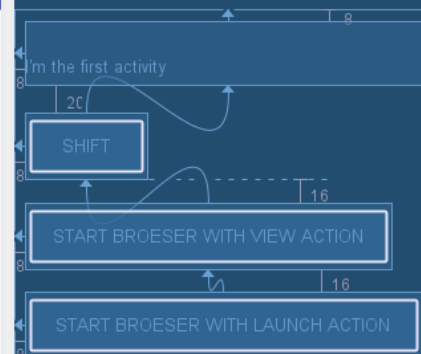
HelloWorld

I'm the first activity

SHIFT

START BROESER WITH VIEW ACTION

START BROESER WITH LAUNCH ACTION

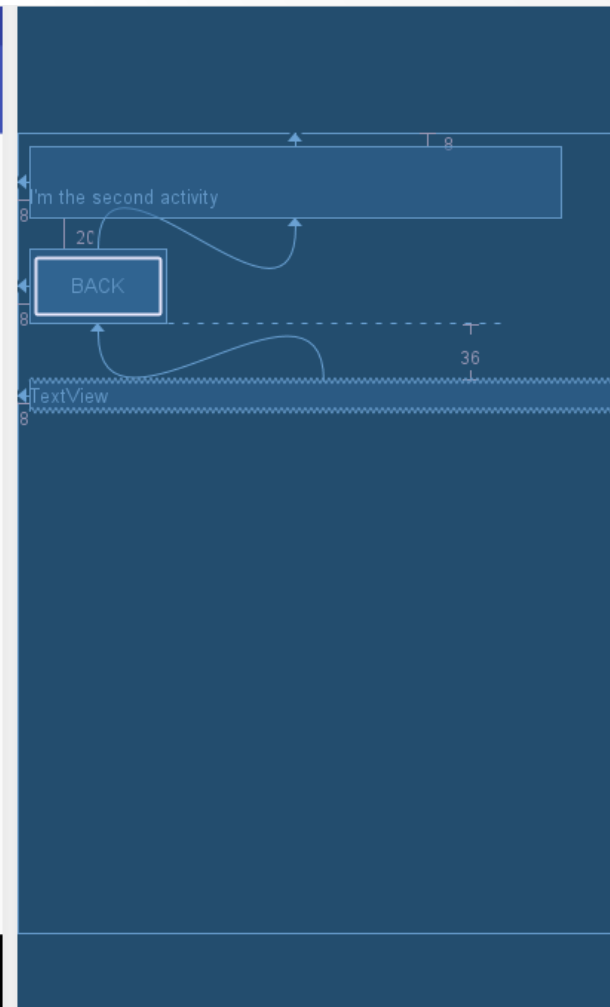
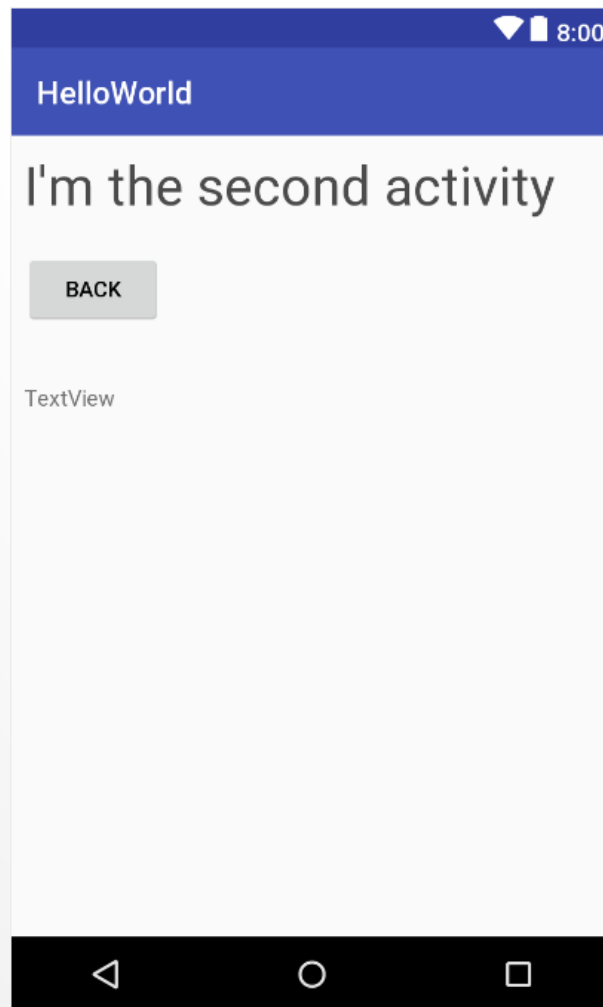


INTENT FILTER

```
<?xml version="1.0" encoding="utf-8"?>
<android.support.constraint.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="com.example.bishop.helloworld.Main2Activity">
    <TextView
        android:id="@+id/Date"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignParentTop="true"
        android:layout_centerHorizontal="true"
        android:layout_marginStart="8dp"
        android:layout_marginTop="8dp"
        android:text="I'm the second activity"
        android:textAppearance="@style/TextAppearance.AppCompat.Display1"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent" />

    <Button
        android:id="@+id/button"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginStart="8dp"
        android:layout_marginTop="20dp"
        android:text="back"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/Date" />

    <TextView
        android:id="@+id/textView1"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:layout_marginStart="8dp"
        android:layout_marginTop="36dp"
        android:text="TextView"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/button" />
</android.support.constraint.ConstraintLayout>
```



INTENT FILTER

```
package com.example.bishop.helloworld;

import android.content.Intent;
import android.net.Uri;
import android.support.v7.app.AppCompatActivity;
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);
        Button b = (Button)findViewById(R.id.button);
        Button b2 = (Button)findViewById(R.id.button2);
        Button b3 = (Button)findViewById(R.id.button3);
        b.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                Intent i = new Intent( packageContext: MainActivity.this, Main2Activity.class);
                startActivity(i);
            }
        });
        b2.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                Intent i = new Intent(Intent.ACTION_VIEW, Uri.parse("http://www.baidu.com"));
                startActivity(i);
            }
        });
        b3.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                Intent i = new Intent( action: "com.example.bishop.helloworld.LAUNCH", Uri.parse("http://www.baidu.com"));
                startActivity(i);
            }
        });
    }
}
```

```
package com.example.bishop.helloworld;

import android.content.Intent;
import android.net.Uri;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.TextView;
import org.w3c.dom.Text;

public class Main2Activity extends AppCompatActivity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main2);
        TextView t = (TextView)findViewById(R.id.textView1);
        Uri url = getIntent().getData();
        t.setText(url.toString());
        Button b = (Button)findViewById(R.id.button);
        b.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                Intent i = new Intent( packageContext: Main2Activity.this, MainActivity.class);
                startActivity(i);
            }
        });
    }
}
```

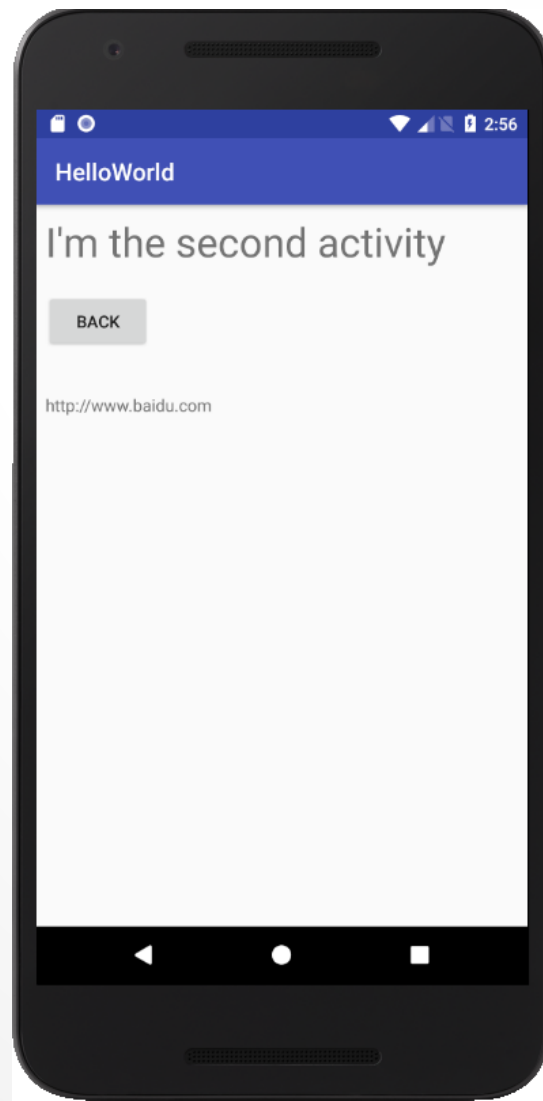
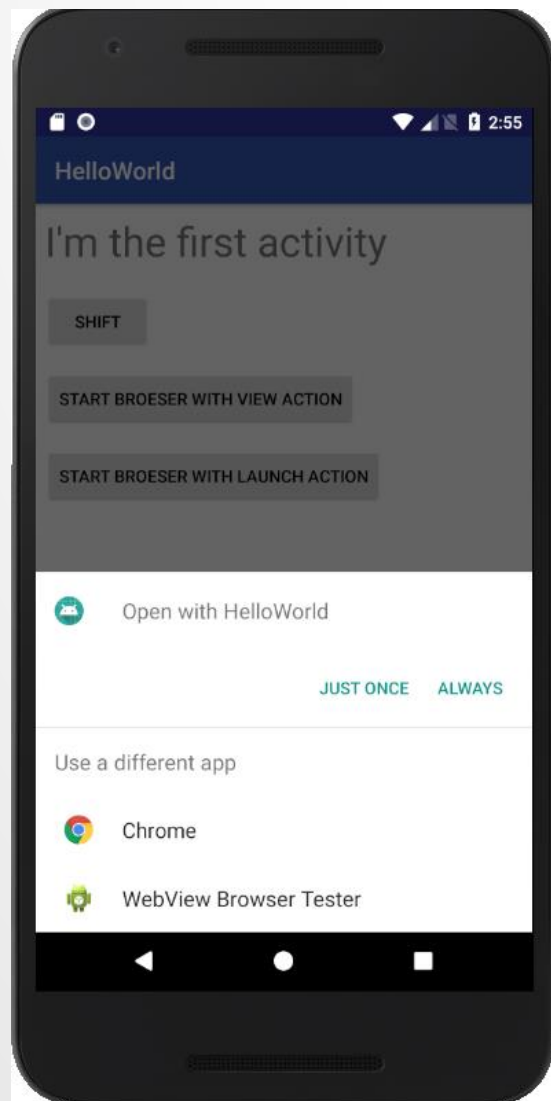
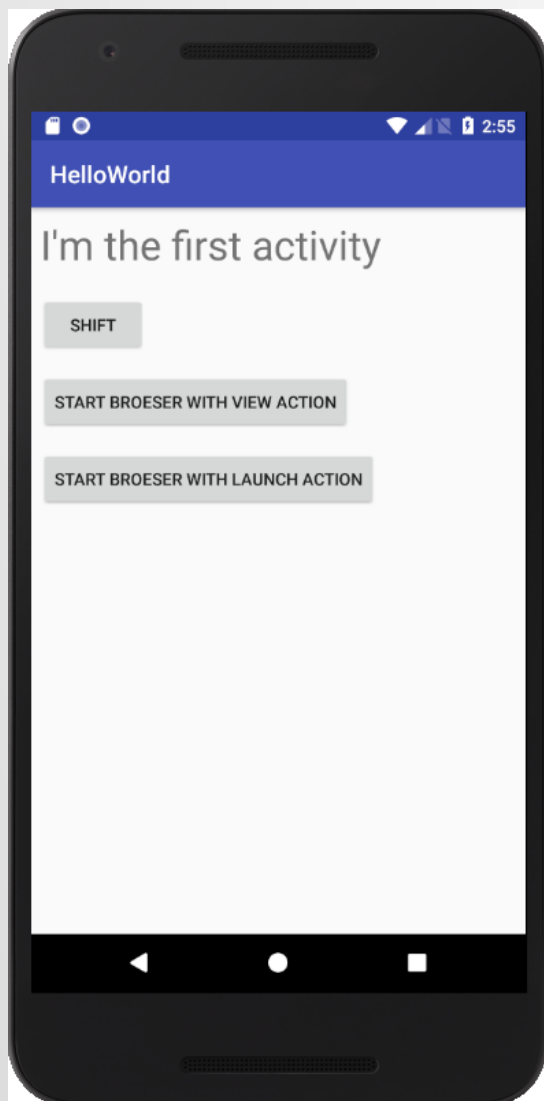
INTENT FILTER

- 注意:
- 课本中AndroidManifest.xml缺少一行
- <category android:name=
- "android.intent.category.BROWSABLE
- "/>

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

    <application
        android:allowBackup="true"
        android:icon="@mipmap/ic_launcher"
        android:label="@string/app_name"
        android:roundIcon="@mipmap/ic_launcher_round"
        android:supportRtl="true"
        android:theme="@style/AppTheme">
        <activity android:name=".MainActivity">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />
                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
        <activity android:name=".Main2Activity">
            <intent-filter>
                <action android:name="android.intent.action.VIEW" />
                <action android:name="com.example.bishop.helloworld.LAUNCH" />
                <category android:name="android.intent.category.DEFAULT" />
                <category android:name="android.intent.category.BROWSABLE"/>
                <data android:scheme="http" />
            </intent-filter>
        </activity>
    </application>
</manifest>
```

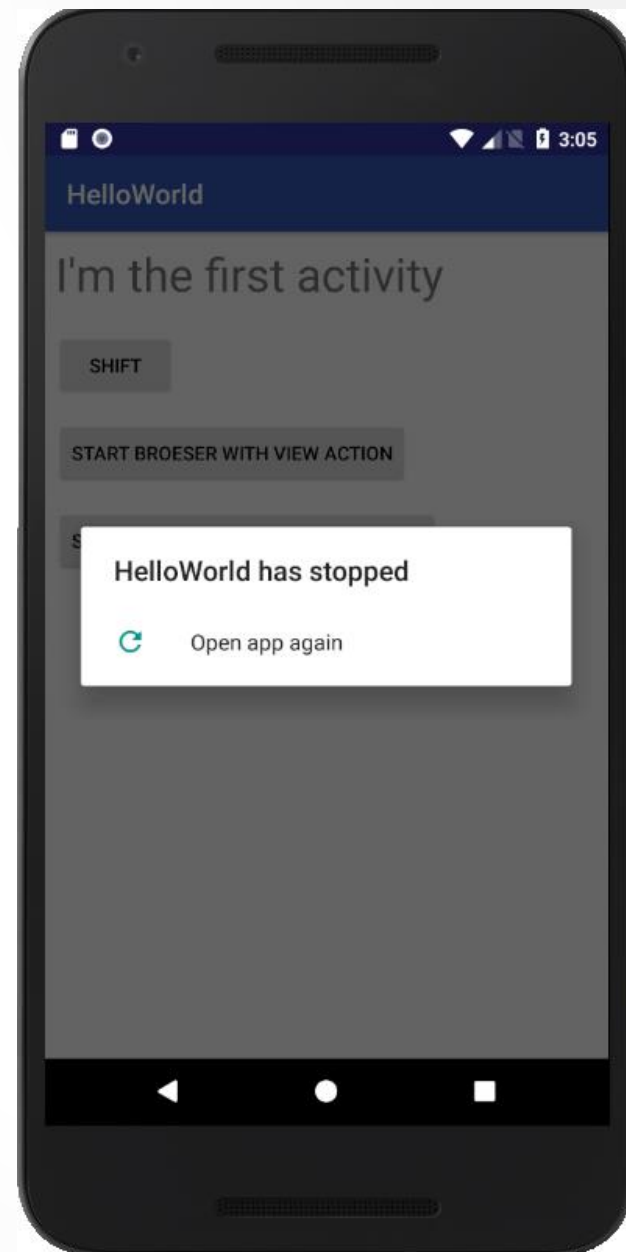

INTENT FILTER



INTENT FILTER

- 修改MainActivity.java中的URI
- 使用https替代http
- 将会导致Android找不到可调用的组件
- 程序异常终止

```
public void onClick(View view) {  
    Intent i = new Intent( action: "com.example.bishop.helloworld.LAUNCH", Uri.parse("https://www.baidu.com"));  
    startActivity(i);  
}
```



THE END.