LEARNING

android-activity

#android-activity

Table of Contents

About

You can share this PDF with anyone you feel could benefit from it, downloaded the latest version from: android-activity

It is an unofficial and free android-activity ebook created for educational purposes. All the content is extracted from Stack Overflow Documentation, which is written by many hardworking individuals at Stack Overflow. It is neither affiliated with Stack Overflow nor official android-activity.

The content is released under Creative Commons BY-SA, and the list of contributors to each chapter are provided in the credits section at the end of this book. Images may be copyright of their respective owners unless otherwise specified. All trademarks and registered trademarks are the property of their respective company owners.

Use the content presented in this book at your own risk; it is not guaranteed to be correct nor accurate, please send your feedback and corrections to info@zzzprojects.com

Chapter 1: Getting started with android-activity

Remarks

This section provides an overview of what android-activity is, and why a developer might want to use it.

It should also mention any large subjects within android-activity, and link out to the related topics. Since the Documentation for android-activity is new, you may need to create initial versions of those related topics.

Examples

Installation or Setup

Detailed instructions on getting android-activity set up or installed.

Activity

Activity is complete screen. UI is **XML** based and

```
package com.example.android.activity;

import android.os.Bundle;
import android.app.Activity;

public class MainActivity extends Activity {

@Override
   public void onCreate(Bundle savedInstanceState) {
      super.onCreate(savedInstanceState);
      setContentView(R.layout.activity_main);

   }
}
```

NOTE Activity must be declared in `AndroidManifest.xml` before using it.

E.g:

```
<activity android:name=".MainActivity">
    <intent-filter>
        <action android:name="android.intent.action.MAIN" />

        <category android:name="android.intent.category.LAUNCHER" />
    </intent-filter>
</activity>
```

Every activity has its layout file in `xml` format, we include its layout using setContentView method of Activity class. E.g. `setContentView(R.layout.activity_main)`

Layout file example

```xml
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:id="@+id/activity_dashboard"
    android:layout_width="match_parent"
    android:layout_height="match_parent">

    //Add other views here

</LinearLayout>
```

Read Getting started with android-activity online: https://riptutorial.com/android-activity/topic/1890/getting-started-with-android-activity

Chapter 2: Hello World

Remarks

`onCreate()` is the most essential part of an activity, it's where most of your activity logic goes.

Examples

Basic Activity structure

Activity is the root `UserInterface` in `Android` and have it's own life-cycle.

MainActivity.java

```
public class MainActivity extends Activity {
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        Toast.makeText(this, "Activity created sucessfully!", Toast.LENGTH_LONG).show();
    }
}
```

AndroidManifest.xml (should be edited)

```
<manifest ... >
    <application ... >
        <activity
            android:name=".MainActivity"
            android:theme="@android:style/Theme.AppCompat">
        </activity>
    </application>
</manifest>
```

Read Hello World online: https://riptutorial.com/android-activity/topic/2868/hello-world

Chapter 3: overriding back button

Examples

Double back to close application

You may have noticed that many applications have double-back-click functionality to exit the app. In this example, we are overriding the default back button action using the `onBackPressed()` method override.

This method will `Toast` a message for the single back-click action, and will close the app if the user clicks the back button twice within two seconds. We use a `Handler` for the functionality that relies on the two-second interval.

```
boolean singleBack = false;

@Override
public void onBackPressed() {
   if (singleBack) {
    super.onBackPressed();
    return;
   }

   this.singleBack = true;
   Toast.makeText(this, "Double Back to exit", Toast.LENGTH_SHORT).show();

   new Handler().postDelayed(new Runnable() {

      @Override
      public void run() {
          singleBack=false;
      }
   }, 2000);
}
```

Read overriding back button online: https://riptutorial.com/android-activity/topic/3812/overriding-back-button

Credits

S. No	Chapters	Contributors
1	Getting started with android-activity	Community, user12345
2	Hello World	Daniel Nugent, George, Nayan Srivastava, yubaraj poudel
3	overriding back button	Daniel Nugent, Prakash Bala