Application Fundamentals & WebView

Agenda

❑ The Location Object in Android

❑ Get the Current Location using Android App

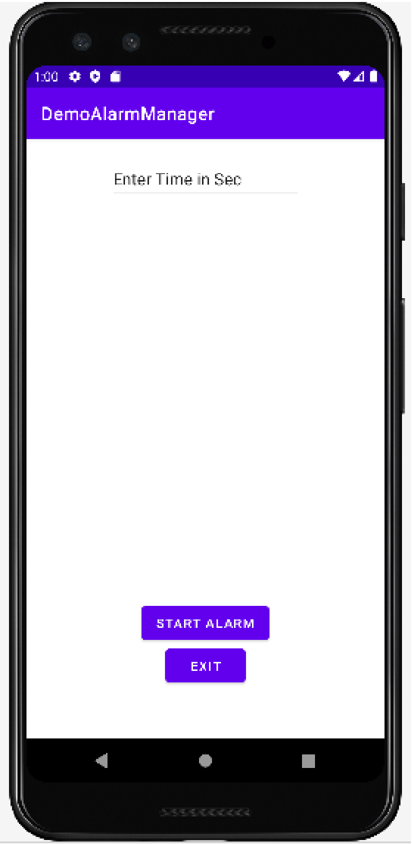
❑ Get the Updated Location using Android App

❑ Location Quality of Service in Android App

AlarmManager in Android

- You may access the system alarm using Android AlarmManager.
- You can programme tasks in your application to run at a specified time in the future with the aid of Android AlarmManager.
- Whether your phone is on or off, it still functions.
- The AlarmManager has a CPU wake lock that ensures the phone won't go to sleep before the broadcast is finished.
- If you restart the device after turning it off, alarms will be erased.

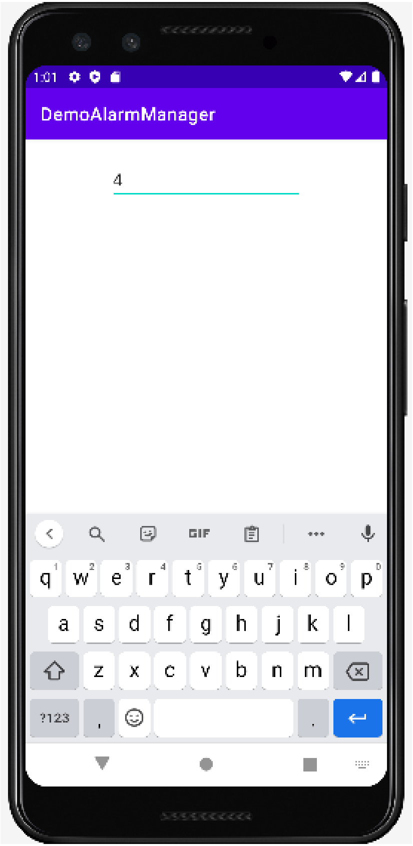
AlarmManager in Android



1:00

DemoAlarmManager

Enter Time in Sec



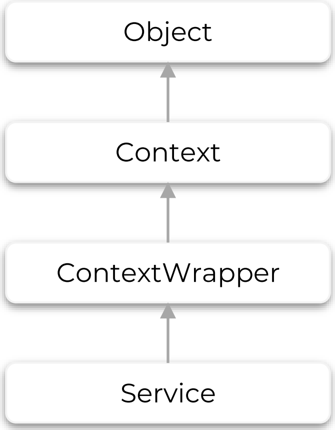
1:01

DemoAlarmManager

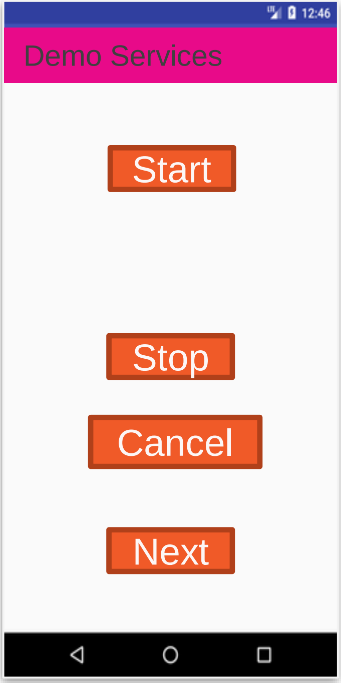
4

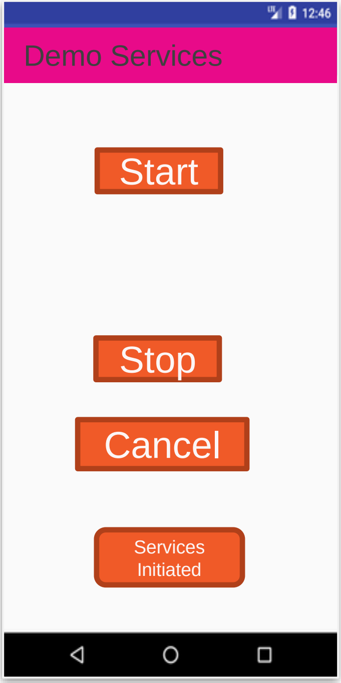
Service in Android App

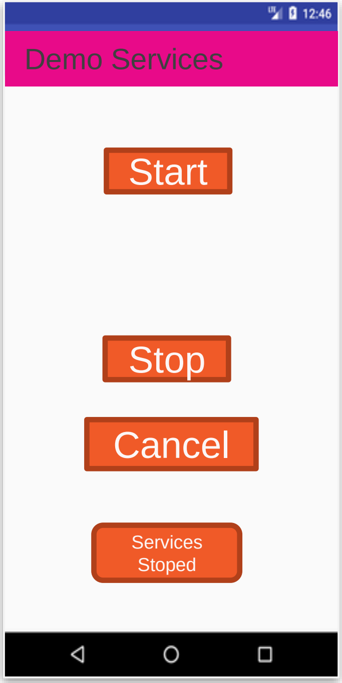
- A component called an Android Service is used to carry out background tasks including playing music, managing network transactions, connecting with content providers, etc.
- There isn't any UI (user interface)
- Even if the application instance is destroyed, the service continues to execute in the background.
- Android application



Service in Android App







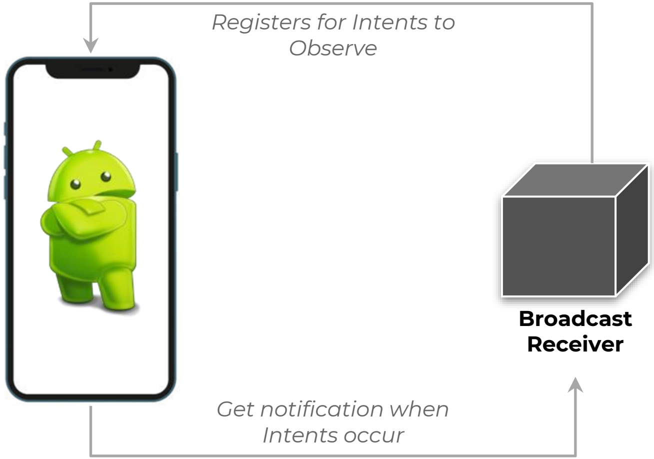
Broadcast Receiver in Android in Android App

- Simply put, broadcast receivers react to broadcast messages sent by other programmers or by the system itself. These messages are occasionally referred to as events or intentions.
- Programs, for instance, might launch broadcasts to inform other applications that certain data has been downloaded to the device and is available for usage.
- This transmission will be intercepted by the broadcast receiver for your application, which will then take the necessary action.

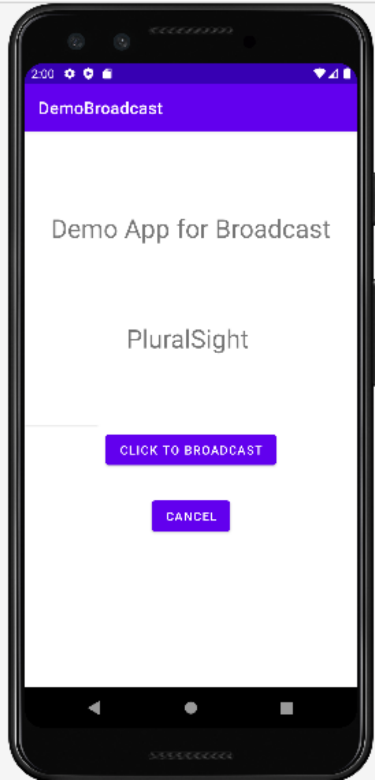
Broadcast Receiver in Android App

To enable Broadcast Receiver to receive system broadcast of intents, the next two procedures are required..

- Create a Broadcast Receiver
- Register a Broadcast Receiver


Registers for Intents to Observe
Broadcast Receiver
Get notification when Intents occur

Broadcast Receiver in Android

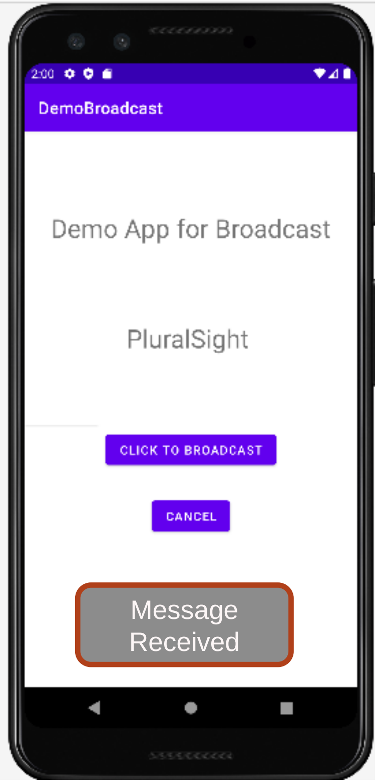


2:00

DemoBroadcast

Demo App for Broadcast

PluralSight



2:00

DemoBroadcast

Demo App for Broadcast

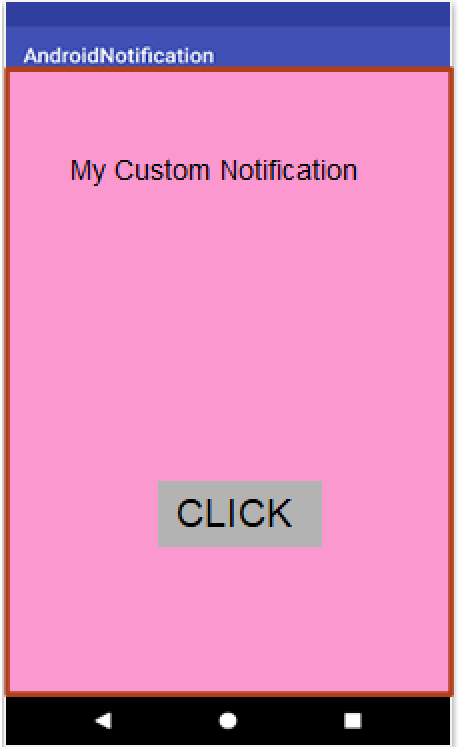
PluralSight

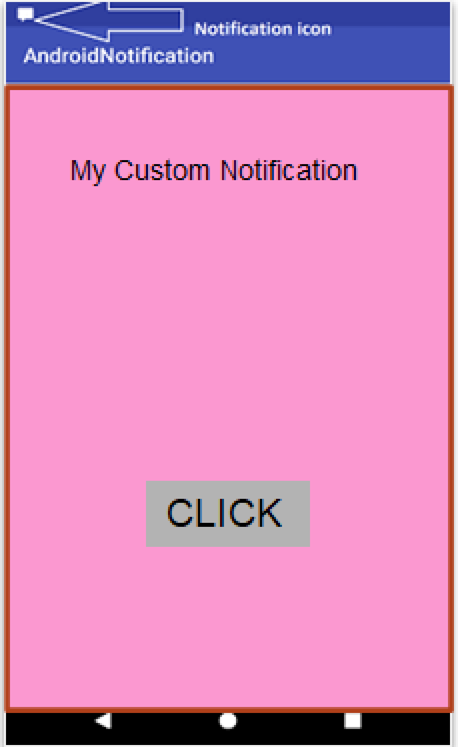
Message Received

Notification Manager in Android

- Even if an application is not running or active, notifications provide concise, timely information on the actions that have occurred in that application.
- Users can understand the notification's icon, title, and brief text content by looking at it.
- The NotificationCompat.Builder object's methods are used to set the properties of Android notifications. Such techniques include following methods:
    - **setSmallIcon()**
    - **setContentTitle()**
    - **setContentText()**
    - **setAutoCancel()**
    - **setPriority()**

Notification Manager in Android





WebView in Android App

- WebView enables your application to display web pages.
- WebView are used to show structured output from an HTML-encoded string.
- Your programme can become a web application using WebView.
- You must include a <WebView> element in your XML layout file in order to add WebView to your application.

WebView in Android

Methods used in WebView are:

canGoBack()

canGoForward()

clearHistory()

destroy()

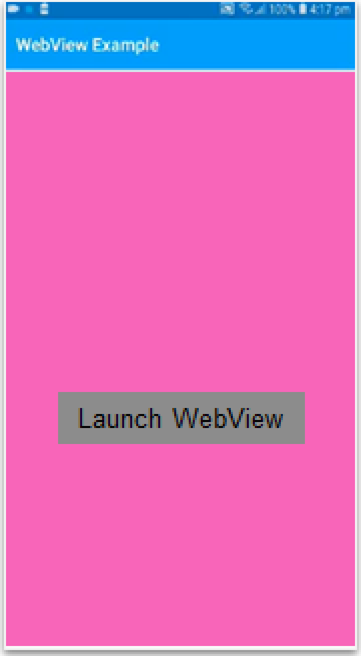
findAllAsync(String find)

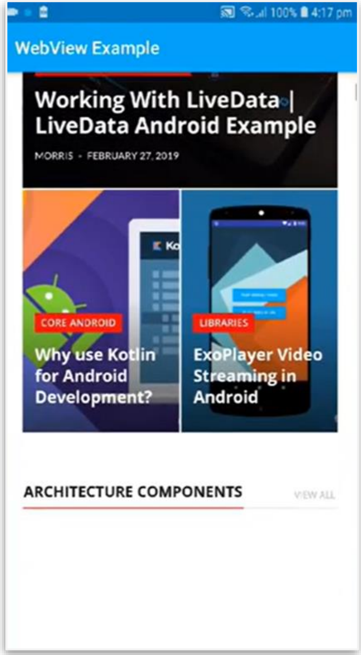
getProgress()

getTitle()

getUrl()

WebView in Android





Working With LiveData | LiveData Android Example

MORRIS · FEBRUARY 27, 2019

CORE ANDROID

Why use Kotlin for Android Development?

LIBRARIES

ExoPlayer Video Streaming in Android

ARCHITECTURE COMPONENTS

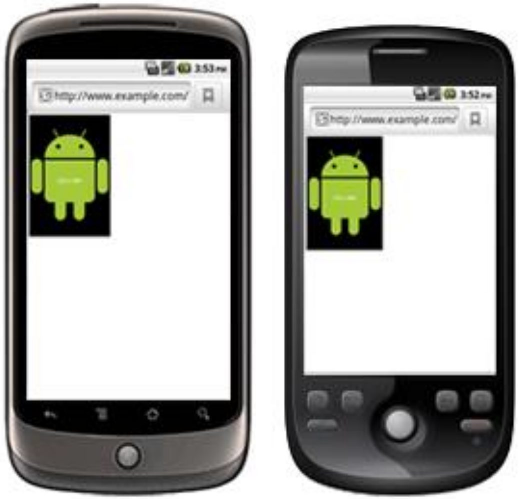
VIEW ALL

Targeted Screens from Web Apps

- Android devices come in a variety of screen sizes and resolutions; therefore you should take the following elements into consideration when designing your web pages for them.
- You should carefully evaluate how your web pages seem on various sorts of screens if you're creating an Android web app or redesigning one for mobile devices.

Targeted Screens from Web Apps

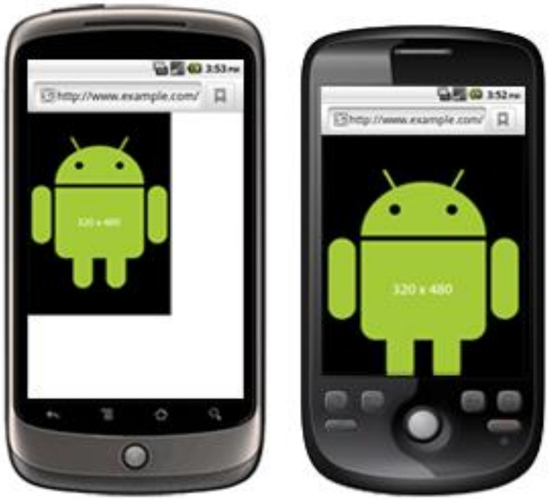
The size of the viewport and scale of the web page



high-density

medium-density

The device's screen density



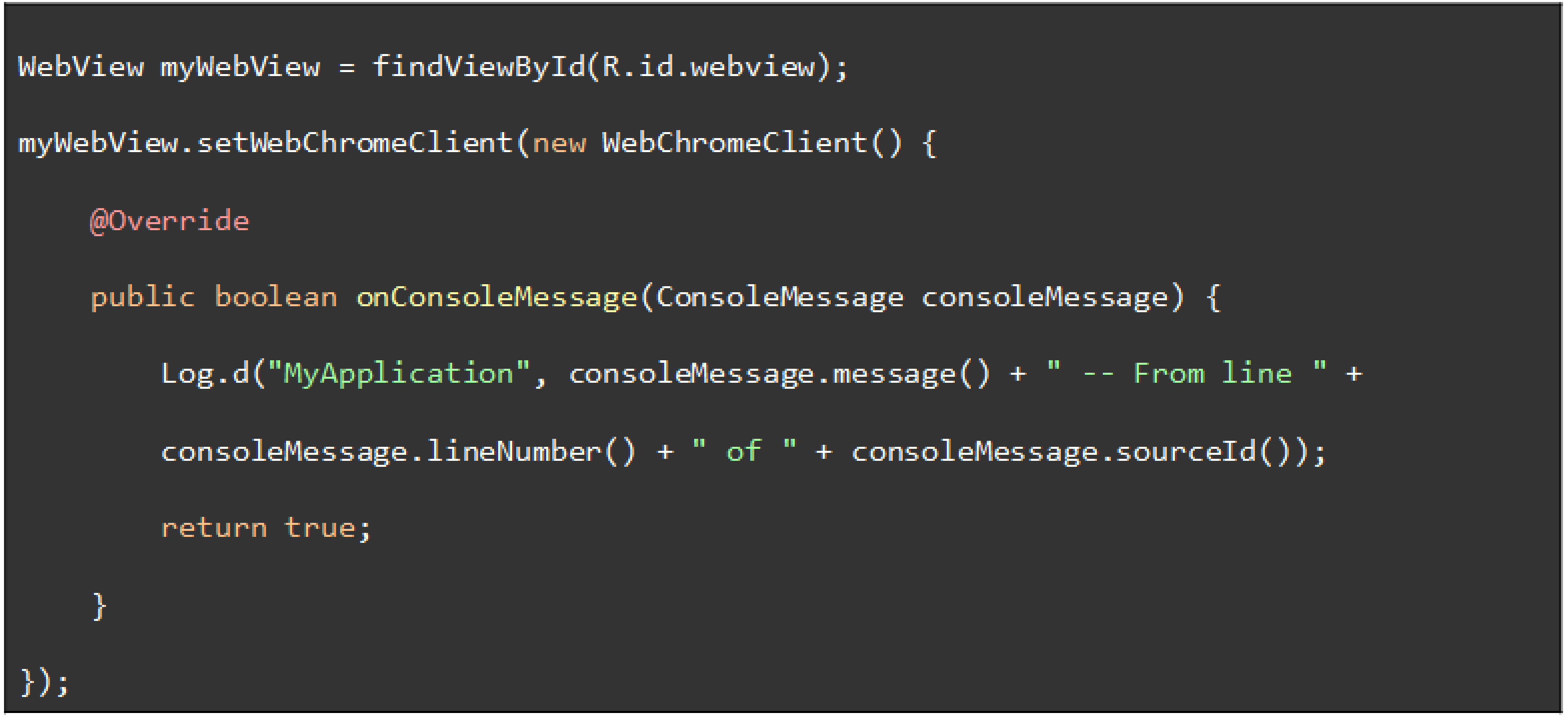
high-density

medium-density

Debugging Web Apps in Android

- Using the console JavaScript APIs, you can debug your JavaScript, and the logcat utility lets you see the output messages.
- If you are comfortable with using Firebug or Web Inspector to debug web pages, you'll find that using console.log() to do so is pretty simple.
- The majority of the same APIs are supported by Android's WebKit framework, allowing you to obtain logs from your web page when debugging in your WebView.

Debugging Web Apps in Android



```
WebView myWebView = findViewById(R.id.webview);

myWebView.setWebChromeClient(new WebChromeClient() {

    @Override

    public boolean onConsoleMessage(ConsoleMessage consoleMessage) {

        Log.d("MyApplication", consoleMessage.message() + " -- From line " +

        consoleMessage.lineNumber() + " of " + consoleMessage.sourceId());

        return true;

    }

});
```

Web Apps Best Practices in Android

- Send mobile users to a mobile-specific version of your website.
- Use a DOCTYPE with proper markup that is suitable for mobile devices.
- Avoid numerous file requests by using viewport meta data to appropriately resize your website.

Questions