Android
Advance
Lesson 3
Intent, File,
Permission



### **Outline**

- Introduction Intents
- II. Work with intents
- III. Working with files
- IV. Permission on device 6.0 and above





### What is intent?

- Intent is an abstract description of an operation to be performed.
- It can be used .with
  - startActivity to launch an Activity,
  - send broadcast to any interested BroadcastReceiver components
  - startService(Intent) or bindService(Intent, ServiceConnection, int)





# Navigate the screen

- Intent using open another Activity from current Activity
- Use putExtra and get<type>Extra to pass data between activities

Constructor/Method	Description
Intent(context, class)	Create an intent for the specific class in the project
putExtra(name, value)	Stores the specified value or array of values with the specified name
getStringExtra(name)	Gets the string value with the specified name
getIntExtra(name, default)	Gets the int value with the specified name. If no int value exits with the specified name, this method return default value



### View a URL in a web browser

Use ACTION\_VIEW to open URL

```
Intent browserIntent = new Intent(Intent.ACTION_VIEW,
Uri.parse("http://www.google.com"));
startActivity(browserIntent);
```



## Dial or call a phone number

> Add CALL\_PHONE permission in the AndroidManifest.xml file

<uses-permission android:name="android.permission.CALL PHONE"></uses-permission>

Create Intent class

```
Intent intentcall = new Intent();
intentcall.setAction(Intent.ACTION_CALL);
intentcall.setData(Uri.parse("tel:" + phonenumber));
startActivity(intentcall);
```



## **Request Image from Camera**

> Add CAMERA permission in the AndroidManifest.xml file

<uses-permission android:name="android.permission.CAMERA"></uses-permission>

Create Intent class

```
static final int REQUEST_IMAGE_CAPTURE = 1;
private void dispatchTakePictureIntent() {
    Intent takePictureIntent = new Intent(MediaStore.ACTION_IMAGE_CAPTURE);
    if (takePictureIntent.resolveActivity(getPackageManager()) != null) {
        startActivityForResult(takePictureIntent, REQUEST_IMAGE_CAPTURE);
    }
}
```



# **Request Image from Gallery**

> Add READ\_EXTERNAL\_STORAGE permission in the AndroidManifest.xml file

<uses-permission android:name="android.permission.READ\_EXTERNAL\_STORAGE"></uses-permission>

Create Intent class

```
static final int SELECT_FILE = 1;
private void dispatchGalleryIntent() {
    Intent intent = new Intent(); intent.setType("image/*");
    intent.setAction(Intent.ACTION_GET_CONTENT);
    startActivityForResult(Intent.createChooser(intent, "Select File"),SELECT_FILE);
}
```





### How to download a file from the Internet

- Use Java and Android APIs to download a file by reading input from the Internet and writing output to the file system
  - Add INTERNET permision

<uses-permission android:name="android.permission.INTERNET"/>

#### Use URL class

Method	Description
openStream()	Return an InputStream object for the specified URL

#### • Save file use Context

	Menthod	Description	
(	openFileOutPut(fileName, mode)	Return a FileOutputStream object for the specified file. If file doesn't exit this menthod creates it. For the second parameter use MODE_PRIVATE to overwrite any exitsting files.	



### **How to Code**

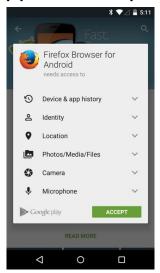
```
final String FILENAME = "news feed.xml";
try{
    // get the input stream
    URL url = new URL("http://rss.cnn.com/rss/cnn tech.rss");
    InputStream in = url.openStream();
    // get the output stream
    FileOutputStream out =
            openFileOutput(FILENAME, Context.MODE_PRIVATE);
    // read input and write output
    byte[] buffer = new byte[1024];
    int bytesRead = in.read(buffer);
    while (bytesRead != -1)
       out.write(buffer, 0, bytesRead);
       bytesRead = in.read(buffer);
    out.close();
    in.close();
catch (IOException e) {
    Log.e("News reader", e.toString());
}
```

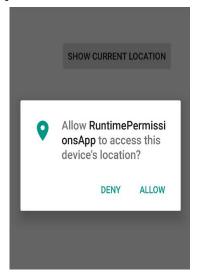




# 1. What is permission

To **protect** the system's integrity and the user's privacy, Android runs each app in a limited access sandbox. If the app wants to use resources or information outside of its sandbox, the app has to explicitly **request permission**.







# 2. Defining Permissions

In AndroidManifest.xml



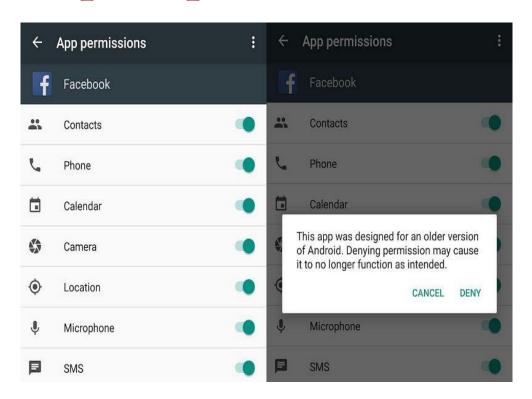
# 3. Dangerous Permission

CALENDAR	READ_CALENDAR
	WRITE_CALENDAR
CAMERA	CAMERA
CONTACTS	READ_CONTACTS
	WRITE_CONTACTS
	GET_ACCOUNTS
LOCATION	ACCESS_FINE_LOCATION
	ACCESS_COARSE_LOCATION
MICROPHONE	RECORD_AUDIO
PHONE	READ_PHONE_STATE
	CALL_PHONE
	READ_CALL_LOG
	WRITE_CALL_LOG
	ADD_VOICEMAIL
	USE_SIP
	PROCESS_OUTGOING_CALLS

SENSORS	BODY_SENSORS
<u>SMS</u>	SEND_SMS
	RECEIVE_SMS
	READ_SMS
	RECEIVE_WAP_PUSH
	RECEIVE_MMS
STORAGE	READ_EXTERNAL_STORAGE
	WRITE_EXTERNAL_STORAGE



## 4. Request permission run time







# Q&A

### **Exercise**

Apply knowledge above do project fetch image from Gallery and display in RecyclerView with GridLayoutManager like photo:

