

UNIT – III : BUILDING BLOCKS OF MOBILE APPS-II

(1) Prepare the definition of Broadcast Receivers?

Broadcast Receivers are the Components that listen for broadcast events & receive/react to the events. These events may be broadcasted by apps or by the system itself. Events such as an incoming call, an incoming SMS, and Wi-Fi availability are some of the common ones generated by the system.

(2) Name some important system events.

Intent.ACTION_BOOT_COMPLETED	- Boot completed.
Intent.ACTION_POWER_CONNECTED	- Power got connected to the device
Intent.ACTION_POWER_DISCONNECTED	- Power got disconnected to the device
Intent.ACTION_BATTERY_LOW	- Battery gets low, typically used to reduce activities in your app which consume power
Intent.ACTION_BATTERY_OKAY	- Battery status good again

(3) Show how Broadcast Receiver is implemented?

A Broadcast Receiver is implemented by extending the BroadcastReceiver class, and overriding its only callback method onReceive().

(4) Generalize the use of Android Telephony APIs?

The Android telephony APIs allows:

- Access the underlying telephone hardware stack
- Create your own dialer
- Integrate call handling and phone state monitoring

(5) Predict the function of SMSManager APIs

SMS sends short text messages between mobile phones. It supports sending both text messages and data messages. Using the SMSManager, we can replace the native SMS application to send text messages, react to incoming texts, or use SMS as a data transport layer.

(6) What is MMS?

MMS (multimedia messaging service) messages have allowed users to send and receive messages that include multimedia attachments such as photos, videos, and audio.

(7) Prepare some information accessed through Telephony Manager.

Through Telephony Manager we can obtain:

- the phone type (GSM or CDMA),
- unique ID (IMEI or MEID),
- software version,
- number.

(8) Highlight the different data persistence methods available in Android?

- Shared Preferences
- Internal Storage
- External Storage
- SQLite Database

(9) What is Shared Preferences?

Applications need a lightweight data storage mechanism for storing application state, configuration options, simple user information, and other user's data. This mechanism on Android platform is called "Shared Preferences" and provides a simple preferences system for storing primitive application data at the Activity level.

The `SharedPreferences` class provides a general framework that allows you to save and retrieve persistent key-value pairs of primitive data types. The preferences are not shared across all of application's activities and it is not possible to share preferences outside of the package.

(10) Show the data types supported by SharedPreferences.

- Boolean values
- Float values
- Integer values
- Long values
- String values

(11) How to read and write persistent key-value pairs of primitive data types?

To write values:

- Call `edit()` to get a `SharedPreferences.Editor`.
- Add values with methods such as `putBoolean()` and `putString()`.
- Commit the new values with `commit()`

To read values, use `SharedPreferences` methods such as `getBoolean()` and `getString()`.

(12) List out the features of Android mobile devices internal and external storage.

Internal storage: Built into the device.

- guaranteed to be present
- typically smaller (~1-4 gb)
- can't be expanded or removed
- specific and private to each app
- wiped out when the app is uninstalled

External storage: Card that is inserted into the device. (such as a MicroSD card)

- can be much larger than internal storage (~8-32 gb)
- can be removed or transferred to another device if needed

- may not be present, depending on the device
- read/writable by other apps and users; not private to your app
- not wiped when the app is uninstalled, except in certain cases

(13) Name some methods for reading and writing files in internal storage.

An activity has methods you can call to read/write files:

- `getFilesDir()`- returns internal directory for your app
- `getCacheDir()`- returns a "temp" directory for scrap files
- `getResources().openRawResource(R.raw.id)` - read an input file from `res/raw/`
- `openFileInput("name", mode)` - opens a file for reading
- `openFileOutput("name", mode)` - opens a file for writing

(14) What is SQLite?

SQLite is a popular, open source, relational database management system (RDBMS) that is widely used in mobile platforms such as Android, Apple iOS, BlackBerry and Symbian. The database is implemented in C and is very light, around 500 KB.

(15) What is an Enterprise Data?

The data over the network to communicate with enterprise systems is called Enterprise data.

(16) What are the permissions require for an app to request data over the network?

Accessing data over the network requires an app to request `android.permission.INTERNET`. The app also needs to request `android.permission.ACCESS_NETWORK_STATE` permission to check network connectivity by accessing network state of the device.

(17) Why RESTful Web services are popular among mobile clients?

Sheer simplicity, light-weight approach, and support for simple CRUD operations have resulted in the popularity of RESTful Web services.