LEZIONE 7 - 01/03/2016

https://github.com/elbuild/corso-android.git

PRINCIPALI COMPONENTI DI ANDROID

- Activity. Componenti che sovrintendono alla gestione della UI e controllano l'interazione con l'utente.
- Fragment. Componenti che consentono un'approccio modulare nella realizzazione e gestione di interfacce utente.
- Intent
- BroadcastReceiver
- Service
- **ContentProvider**

STORAGE IN ANDROID

STORAGE ANDROID

- Ci sono diversi modi di immagazzinare localmente informazioni in maniera persistente:
  - SharedPreferences
  - Internal Storage
  - External Storage
  - SQLite database
  - Network clients with local cache (HTTP Expiry and/or Etag)

SHARED PREFERENCES

STORAGE ANDROID

- Le SharedPreferences consentono di salvare numerosi tipi, primitivi e non in uno storage chiave/valore

- Si possono salvare int, double, float, boolean, long e stringhe. Grazie alle stringhe è possibile salvare e recuperare POJO serializzandoli JSON.

- Ogni applicazione ha la sua istanza di storage SharedPreferences, o anche più di una, a cui accede passando un nome al metodo getSharedPreferences() esposto dal Context

- Nell'accedere alle SharedPreferences è possibile specificare un mode fra MODE_PRIVATE (default), MODE_WORLD_READABLE, MODE_WORLD_WRITABLE ed altri.

- Scrivere sulle SharedPreferences

```
Editor editor = getSharedPreferences("name", MODE_PRIVATE).edit();
editor.putString("key", "value");
editor.commit();
```

- Ovviamente ci sono metodi put per ogni tipo supportato.
- Altre operazioni possibili sono clear() e remove(key).
- Il valore scritto sopravvive al reboot della app e del device, ma non al "Cancella dati" fatto dai Settings Android.

- Leggere dalle SharedPreferences

```
SharedPreferences shared = getSharedPreferences("name",
MODE_PRIVATE).edit();
String value = shared.getString("key");
```

- Ovviamente ci sono metodi get per ogni tipo supportato.
- Il metodo in get supporta anche un secondo parametro che è il valore da ritornare di default nel caso non ci sia un mapping per la chiave.
- L'utilizzo delle SharedPreferences è adeguato per casi in cui i dati siano limitati, schema free, e su cui non servano operazioni di ricerca o filtering.

INTERNAL STORAGE

STORAGE ANDROID

- L'INTERNAL STORAGE è uno spazio virtuale sulla memoria persistente del device ad uso esclusivo dell'applicazione. Le altre applicazioni non lo possono leggere/scrivere e nemmeno l'utente (se non ha root ovviamente...)

- Le operazioni di salvataggio sono quelle proprie di un file e l'accesso in lettura scrittura è quello che si avrebbe su un comune file system.

- Le operazioni di scrittura/lettura si appoggiano sulle primitive JDK per la manipolazione dei file basata su stream (FileOutputStream e InputStream).

- Questo tipo di storage è indicato per cache locali, e dati schema free anche di grandi dimensioni (es. immagini, PDF, etc) che hanno bisogno di essere accessibili in tempi rapidi.

- Scrivere sull'InternalStorage:

```
String FILENAME = "hello_file.txt";
String string = "hello world!";

FileOutputStream fos = openFileOutput(FILENAME, Context.MODE_PRIVATE);
fos.write(string.getBytes());
fos.close();
```

- Leggere dall'InternalStorage:

```
FileInputStream in = openFileInput("filename.txt");
InputStreamReader inputStreamReader = new InputStreamReader(in);
BufferedReader bufferedReader = new BufferedReader(inputStreamReader);
StringBuilder sb = new StringBuilder();

String line;

while ((line = bufferedReader.readLine()) != null) {
    sb.append(line);
}

in.close();
```

STORAGE ANDROID

- L'InternalStorage ha alcune directory predefinite come la directory CACHE, accessibile con *getCacheDir()*

- I contenuti della directory cache sono "meno persistenti" perchè l'OS può cancellarli se lo spazio di archiviazione termina e anche perchè c'è un controllo specifico nei settings che permette all'utente di cancellare la cache.

- Esistono altri metodi utili come getFilesDir() che consente di accedere la radice della directory dei files.

- Altri metodi consentono di creare cartelle, creare e cancellare files, proprio come su un FS.

EXTERNAL STORAGE

STORAGE ANDROID

- L'ExternalStorage è una periferica virtuale disponibile su ogni device Android. Potrebbe trattarsi della SD Card ma anche di una parte della memoria di massa.

- Le operazioni di salvataggio sono quelle proprie di un file e l'accesso in lettura scrittura è quello che si avrebbe su un comune file system. L'utente e le altre applicazioni possono accedere allo stesso storage.

- Le operazioni di scrittura/lettura sono disponibili condizionalmente alla concessione di una permission.

```
<manifest ...>
    <uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE" />
    ...
</manifest>
```

STORAGE ANDROID

- Le operazioni sono simili a quelle della scrittura/lettura da InternalStorage
- I file, a meno non siano definiti privati dalla app, persistono alla elminazione della app stessa.
- Esistono una serie di funzioni helper che danno accesso diretto alle cartelle Download, Immagini, e a tutte le cartelle pubbliche definite dall'OS
- Use case tipici di salvataggio su ExternalStorage sono la condivisione con altre applicazioni o la creazione di DB pubblici (SQLite) da esporre tramite un ContentProvider

IntentServiceExample

https://github.com/elbuild/corso-android.git

Q&A

https://github.com/elbuild/corso-android.git