



Trường ĐH Khoa Học Tự Nhiên Tp. Hồ Chí Minh  
**TRUNG TÂM TIN HỌC**

# Lập trình Android

## Bài 14: *File & SharedPreferences*

Phòng LT & Mạng

<http://csc.edu.vn/lap-trinh-va-csdl>





# Nội dung

---

1. SharedPreferences
2. File



# SharedPreferences

---

- **SharedPreferences** là một cách hiệu quả để lưu dữ liệu.
- Dữ liệu được lưu dưới dạng một cặp **<Key, Value>**, trong đó **Key** là một **string**, còn **Value** là các kiểu dữ liệu cơ bản (int, long, double, boolean, string...)
- Dữ liệu được lưu trong bộ nhớ chính của thiết bị
- Thường dùng để lưu các thông tin trạng thái và chia sẻ giữa các activity trong cùng một ứng dụng

Key	Value



# SharedPreferences – Ví dụ

- Lưu một giá trị string:

```
public void savePreferences(String key, String value) {  
    SharedPreferences p = getSharedPreferences("caches", Context.MODE_PRIVATE);  
    SharedPreferences.Editor edCaches = p.edit();  
    edCaches.putString(key, value);  
    edCaches.commit();  
}
```

Tên file XML chứa dữ liệu SharedPreferences

- Đọc một giá trị string:

```
public String loadPreferences(String key) {  
    SharedPreferences p = getSharedPreferences("caches", Context.MODE_PRIVATE);  
    return p.getString(key, null);  
}
```



# SharedPreferences – Ví dụ

---

- Xoá một giá trị:

```
public void removePreferences(String key) {  
    SharedPreferences p = getSharedPreferences("caches", Context.MODE_PRIVATE);  
    SharedPreferences.Editor edCaches = p.edit();  
    edCaches.remove(key);  
    edCaches.commit();  
}
```

- Xoá tất cả các giá trị:

```
public void clearPreferences() {  
    SharedPreferences p = getSharedPreferences("caches", Context.MODE_PRIVATE);  
    SharedPreferences.Editor edCaches = p.edit();  
    edCaches.clear();  
    edCaches.commit();  
}
```

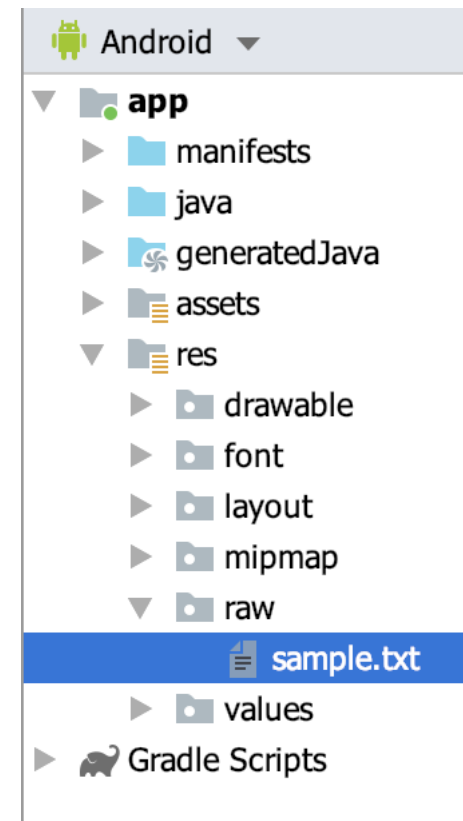


# File

- Đọc Raw File:

```
public static String readRawFile(Context context, int fileResource) {  
    StringBuffer buf = new StringBuffer();  
    InputStream is = context.getResources().openRawResource(fileResource);  
    BufferedReader reader = new BufferedReader(new InputStreamReader(is));  
    try {  
        String str;  
        while (((str = reader.readLine()) != null))  
            buf.append(str + "\n");  
        reader.close();  
        is.close();  
    } catch (IOException e) {  
        e.printStackTrace();  
    }  
    return buf.toString();  
}
```

R.raw.sample

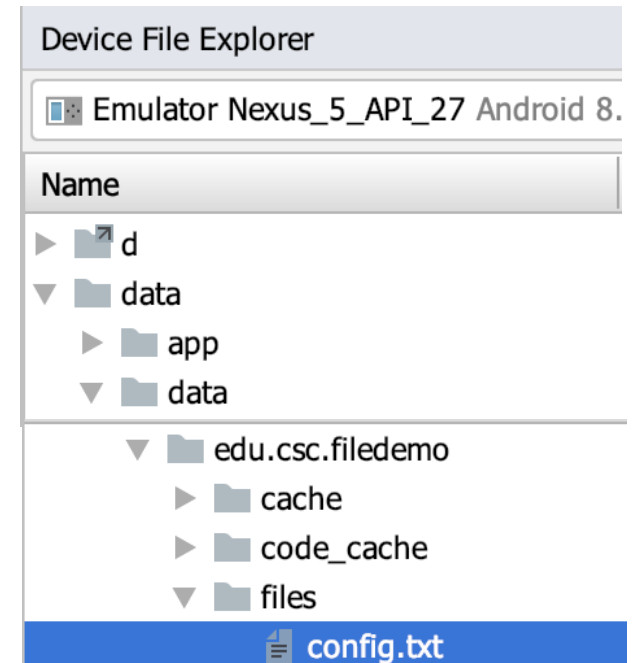




# File

- Ghi Internal File: Việc đọc/ghi file trên Android kế thừa lại từ Java API

```
public static void write(Context context, String fileName, String message) {  
    File file = new File(context.getFilesDir(), fileName);  
    try {  
        FileOutputStream fos = new FileOutputStream(file);  
        OutputStreamWriter osw = new OutputStreamWriter(fos);  
        BufferedWriter writer = new BufferedWriter(osw);  
        writer.write(message);  
        writer.close();  
        osw.close();  
        fos.close();  
    } catch (IOException e) {  
        e.printStackTrace();  
    }  
}
```





# File (tiếp)

---

- Đọc Internal File:

```
public static String read(Context context, String fileName) {
    StringBuffer buf = new StringBuffer();
    try {
        FileInputStream fis = context.openFileInput(fileName);
        if (fis != null) {
            InputStreamReader isr = new InputStreamReader(fis);
            BufferedReader reader = new BufferedReader(isr);
            String str;
            while (((str = reader.readLine()) != null))
                buf.append(str + "\n");
            reader.close();
            isr.close();
            fis.close();
        }
    } catch (IOException e) {
        e.printStackTrace();
    }
    return buf.toString();
}
```





# File (tiếp)

---

- Ghi External File:

```
public static void writeExternalFile(String fileName, String message) {  
    String filePath = Environment.getExternalStorageDirectory().getPath() + "/" + fileName;  
    try {  
        PrintWriter writer = new PrintWriter( new FileWriter(filePath) );  
        writer.println(message);  
        writer.close();  
    } catch (IOException e) {  
        e.printStackTrace();  
    }  
}
```

- Trước khi đọc/ghi **external** file, cần phải khai báo **permissions** tương ứng trong **AndroidManifest.xml** và **request runtime permissions** đối với người dùng

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

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





# File (tiếp)

- Đọc External File:

```
public static String readExternalFile(String fileName) {  
    String filePath = Environment.getExternalStorageDirectory().getPath() + "/" + fileName;  
    String s = "";  
    try {  
        Scanner scanner = new Scanner(new FileReader(filePath));  
        while(scanner.hasNextLine()) {  
            s += scanner.nextLine() + "\n";  
        }  
        scanner.close();  
    } catch (FileNotFoundException e) {  
        e.printStackTrace();  
    }  
    return s;  
}
```

