

Android

Permanent Data Storage

SharedPreferences

- เอาไว้สำหรับเก็บค่าข้อมูลที่ใช้ร่วมกันใน application
- รูปแบบในการเก็บจะอยู่ในรูปแบบ key-value pair
- ตัวอย่าง

```
SharedPreferences sp = getSharedPreferences("HEALTHY", MODE_PRIVATE);  
sp.edit().putString("weight", "84").apply();  
sp.commit();  
Log.d("WEIGHT", sp.getString("weight", "0"));
```

Internal Storage

- ต้อง import java.io มาด้วย

```
try{
    String filename = "test.txt";
    FileOutputStream fos = openFileOutput(filename, MODE_PRIVATE);
    fos.write("data in ".getBytes());
    fos.close();

    FileInputStream fis = openFileInput(filename);
    IOUtils.readInputStreamFully(fis).toString();
}catch(FileNotFoundException e){
    e.printStackTrace();
}catch (IOException ioe){
    ioe.printStackTrace();
}
```

SQLite Database

Every Android app can create and make use of SQLite databases to store large amounts of structured data. As you might already know, SQLite is not only light-weight, but also very fast. If you have experience working with relational database management systems and are familiar with both SQL, which is short for Structured Query Language, and JDBC, which is short for Java Database Connectivity, this might be your preferred storage option.

Create or Open Database

- คำสั่งสำหรับเรียกใช้งานหรือสร้าง database ใหม่ โดยที่ my.db คือชื่อ database ที่เราต้องการสร้าง

```
SQLiteDatabase myDB = openOrCreateDatabase("my.db", MODE_PRIVATE, null);
```

- เราสามารถสร้าง table ได้ด้วยคำสั่ง create table if not exists

```
myDB.execSQL(  
    "CREATE TABLE IF NOT EXISTS user (_id INTEGER PRIMARY KEY AUTOINCREMENT, name VARCHAR(200), age  
    INT, is_single INT)"  
);
```

- เราสามารถ open database ใน fragment class ได้ด้วยคำสั่งดังนี้

```
SQLiteDatabase db = getActivity().openOrCreateDatabase("my.db", MODE_PRIVATE, null);
```

ContentValues

- เอาไว้ประกาศ object ที่ map กับโครงสร้างของ row ใน table ที่เราต้องการ
- โดยที่ parameter แรกคือชื่อ field parameter ที่สอง คือ value

```
ContentValues row1 = new ContentValues();  
row1.put("name", "Alice");  
row1.put("age", 25);  
row1.put("is_single", 1);
```

CRUD

- คำสั่งที่ใช้ในการ insert ใช้ method ผ่าน db ได้ โดย user คือ ชื่อ table , row1 คือ instance ของ ContentValues

```
myDB.insert("user", null, row1);
```

- คำสั่งที่ใช้ในการ update ใช้ method ผ่าน db ได้ โดย user คือ ชื่อ table , row1 คือ instance ของ ContentValues และ _id คือ primary key ของ table user ตอน update ต้องส่ง key เข้าไปด้วย

```
myDB.update("user", row1, "_id="+id, null);
```

- คำสั่งที่ใช้ในการ delete ใช้ method ผ่าน db ได้ โดย user คือ ชื่อ table และ _id คือ primary key ของ table user ตอน delete ต้องส่ง key เข้าไปด้วย

```
myDB.delete("user", "_id="+id, null);
```

Query

- คำสั่งที่ใช้ในการ ดึงข้อมูล ใช้ method ผ่าน db ได้ โดย สามารถระบุ sql statement เข้าไปได้เลย

```
Cursor myCursor =  
    myDB.rawQuery("select name, age, is_single from user", null);  
  
while(myCursor.moveToNext()) {  
    String name = myCursor.getString(0);  
    int age = myCursor.getInt(1);  
    boolean isSingle = (myCursor.getInt(2)) == 1 ? true:false;  
}  
  
myCursor.close;
```


SEND DATA TO FRAGMENT

- STEP Fragment ต้นทาง
 - สร้าง Bundle Class ที่ fragment ต้นทาง
 - set value เข้า class bundle ด้วยคำสั่งดังนี้
 - `bundle.putString("key", "value");`
 - New Fragment Instance
 - Pass bundle to fragment ด้วยคำสั่ง `.setArguments(bundle)`
- STEP Fragment ปลายทาง
 - Get bundle จากต้นทางด้วยคำสั่งดังนี้
 - `getArguments();` จะได้ class bundle ออกมา
 - ดึงค่าจาก bundle ด้วยคำสั่งดังนี้
 - `bundle.getString("key")`

Rest API

- okhttp คือ lib สำหรับให้ android สามารถเรียกใช้ api ต่างๆ ผ่าน http method ได้
- เราจำเป็นต้อง เพิ่ม Gradle lib ดังนี้เข้าไปที่ file build.gradle

implementation 'com.squareup.okhttp3:okhttp:3.11.0'

```
OkHttpClient client = new OkHttpClient();

String run(String url) throws IOException {
    Request request = new Request.Builder()
        .url(url)
        .build();

    Response response = client.newCall(request).execute();
    return response.body().string();
}
```

Get From Server

```
public static final MediaType JSON
    = MediaType.parse("application/json; charset=utf-8");

OkHttpClient client = new OkHttpClient();

String post(String url, String json) throws IOException {
    RequestBody body = RequestBody.create(JSON, json);
    Request request = new Request.Builder()
        .url(url)
        .post(body)
        .build();
    Response response = client.newCall(request).execute();
    return response.body().string();
}
```

Post To Server

Workshop #1

- เมื่อกดที่ menu sleep แล้วให้เปิดหน้า sleep ขึ้นมา
- เพิ่มปุ่ม back ที่หน้า sleep เอาไว้บนสุด เมื่อกดแล้วให้กลับไปหน้า menu
- ส่งงาน 5 Nov เกือบ 10 คะแนน

Workshop #2

Sleep

- แสดงรายการเวลานอนในแต่ละวัน
โดยให้ดึงข้อมูลมาจาก SQLite
- เมื่อกดปุ่ม Add Sleep ให้ไปหน้า
sleep_form
- เมื่อกดที่รายการให้ไปที่
sleep_form โดย load ข้อมูลมา
เพื่อให้แก้ไข

Add Sleep

19-Jan-18

22:00 - 05:30

07:30

18-Jan-18

20:00 - 05:30

09:30

17-Jan-18

20:00 - 05:30

09:30

16-Jan-18

20:00 - 05:30

09:30

Workshop 3

sleep_form

- เมื่อกดปุ่ม Back ให้กลับไปหน้า sleep
- เมื่อกด บันทึกให้ เก็บข้อมูลลง SQLite พร้อมทั้งแสดง message

Back

วันที่

เวลาเข้านอน

เวลาตื่น

Save

ประกาศ

- เสาร์ที่ 17, อาทิตย์ที่ 18 เรียนเวลา 10:00 - 12:00, 13:00 - 17:00
- เป็นวัน present project เก็บคะแนน 40 คะแนน

เงื่อนไขการให้คะแนน Project			
ข้อ	รายละเอียด	คะแนน	หมายเหตุ
1	หน้าจอ Feature List	5	แสดง Feature หลักๆ ของ applicatio
2	หน้าจอสำหรับการ Login	5	เอาไว้ให้ user login เข้าระบบใช้วิธีการใดก็ได้ ดูที่ความปลอดภัย
3	หน้าจอสำหรับการ Register	5	เอาไว้ให้ user ลงทะเบียนเข้าใช้ ระบบ ใช้วิธีการใดก็ได้ ดูที่ความสะดวก ปลอดภัย
4	มีการใช้ กล้องถ่ายรูป หรือ Photos หรือ Images จาก Device	5	การใช้ function พวกนี้ต้องเกี่ยวข้องกับ application ที่ทำออกมาถึงได้คะแนน ถ้าใช้โดยไม่มี ความเกี่ยวข้องกับเรื่องที่ทำจะไม่ได้คะแนน
5	มีการใช้ Map API	5	
6	มีการเก็บข้อมูลบน Firebase	5	
7	มีการเรียกใช้ REST API	5	
8	มีการเก็บข้อมูลลง Device (SQLite, Sharedperference, File)	5	เน้นการ present ถึงคุณประโยชน์ของ application กลุ่มผู้ใช้เป็นใคร ใช้แล้วได้อะไร ทำไมต้องใช้ เป็นต้น
9	ประโยชน์ของ Application ต่อผู้ใช้งาน	5	
10	Mood and Tone มีความกลมกลืนและเข้ากับ Theme ของ application	5	ดูที่การ design ของ application ว่าเข้ากับเรื่อง อาหารไหม icon, รูปภาพ ปุ่ม ux ui เหมาะสมกับกลุ่ม user ที่จะใช้ application ใหม

เงื่อนไขการให้คะแนน

Project

	รายชื่อกลุ่ม	หมายเหตุ	วัน	เวลา
1	วิธีการทำอาหาร		17-Nov	10:00 - 11:00
2	จองร้านอาหาร	จองโต๊ะอาหาร		11:00 - 12:00
3	Local Food Delivery			13:00 - 14:00
4	ค้นหาร้านอาหารบุฟเฟ่ต์			14:00 - 15:00
5	รีวิวร้านอาหาร			15:00 - 16:00
6	Calcurie (แคลคูลี่)	บันทึกข้อมูลการกินอาหาร เพื่อช่วยในการคำนวณแคลอรี่		16:00 - 17:00
7	Food Direction	ระบบชี้้นำการเลือกทานอาหาร	18-Nov	10:00 - 11:00
8	Ingredients finder			11:00 - 12:00
9	FoodBook			13:00 - 14:00
10				14:00 - 15:00
11	AR Cooking	เดี่ยว		15:00 - 16:00

รายชื่อกลุ่ม

และวันเวลาที่ส่ง project

เงื่อนไข

- Application ห้ามมีอาหารดับในระหว่างการ present
- ตอน present สมาชิกคนไหนไม่มา สมาชิกคนนั้นจะได้คะแนนครึ่งหนึ่งของคะแนนกลุ่ม เช่น ทั้งกลุ่มได้ 40 สมาชิกคนนั้นจะได้ 20
- ก่อน present ต้องส่ง code ก่อน
- หลังจากได้ code แล้ว อ. จะเป็นคน build code แล้วให้นักศึกษา present
- การ present จะมีหรือไม่มีเอกสารในการนำเสนอก็ได้ แต่ให้สามารถคนในห้องเข้าใจได้ว่าโปรแกรมมี function อะไรแล้วมีประโยชน์ อย่างไร