

LAB 9 การสร้างแอปพลิเคชันเชื่อมต่อกับ MySQL 2

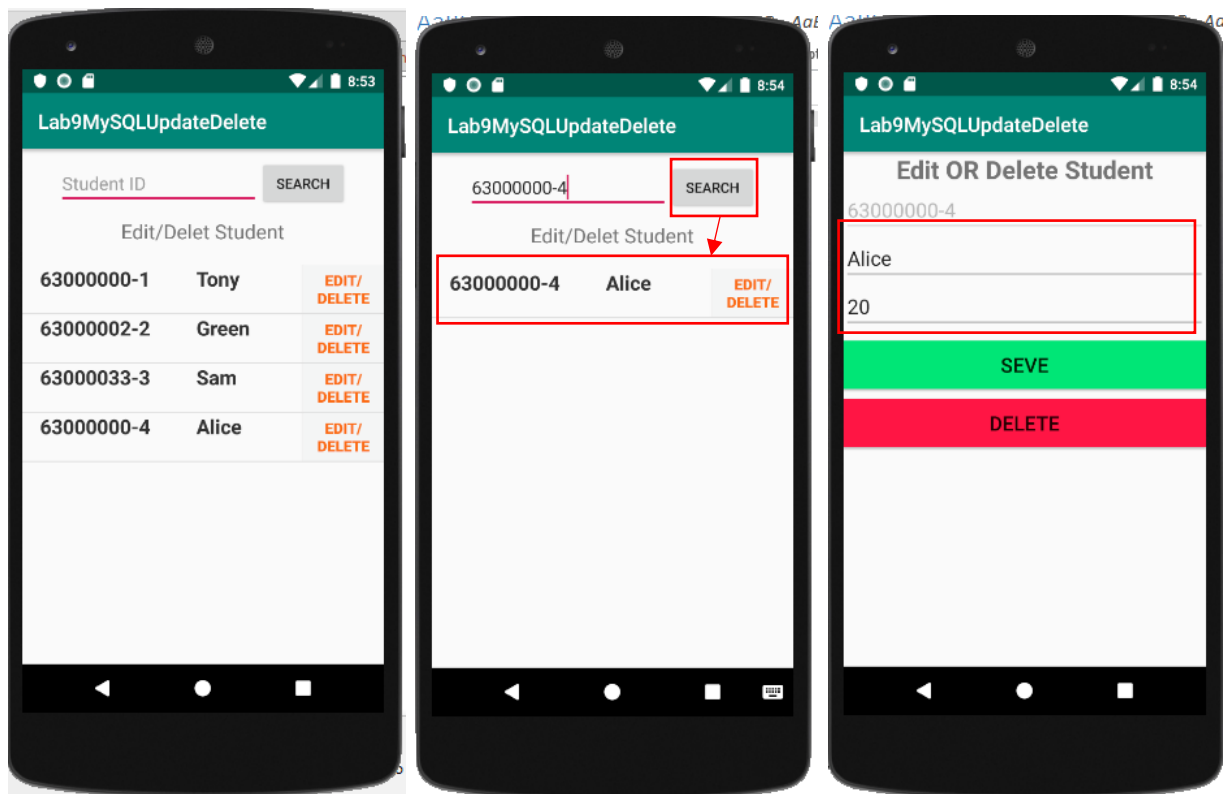
เอกสารประกอบการปฏิบัติการรายวิชา 342267 MOBILE DEVICE PROGRAMMING

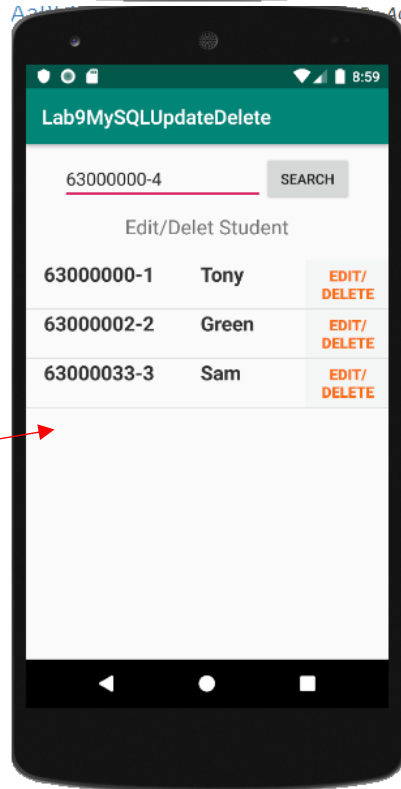
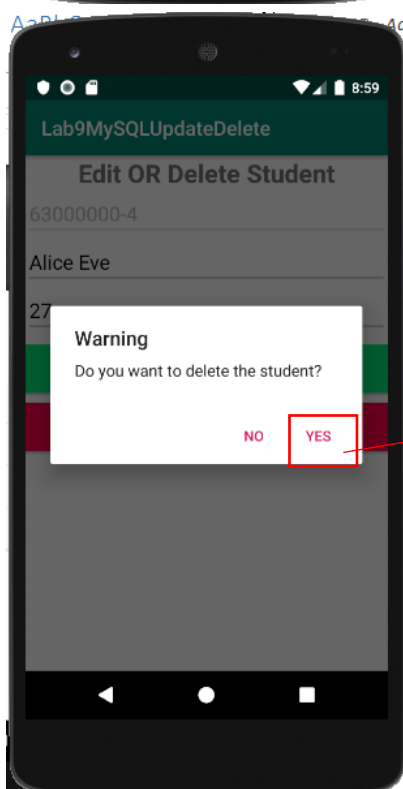
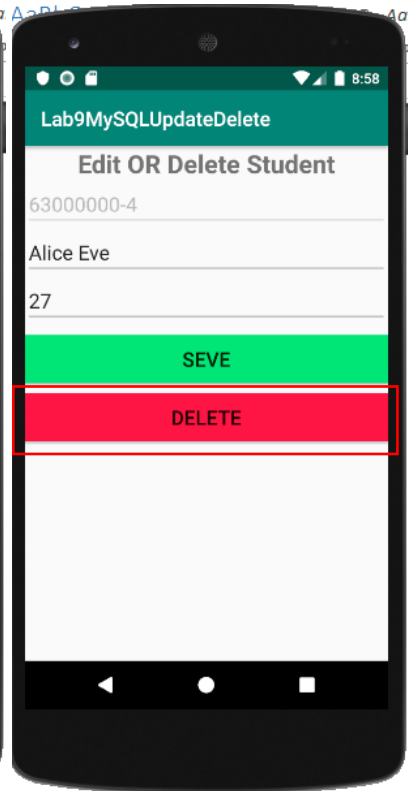
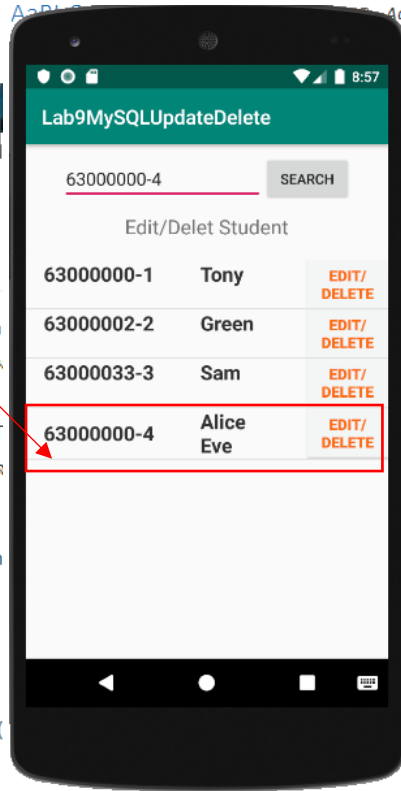
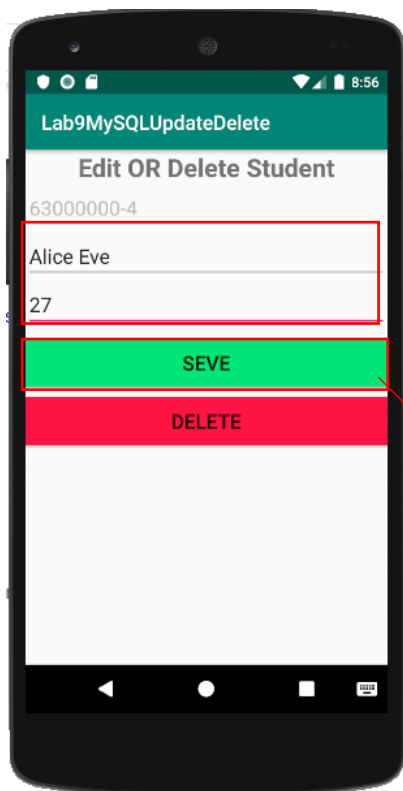
วัตถุประสงค์

1. เพื่อให้นักศึกษาสามารถพัฒนาโปรแกรมให้สามารถเชื่อมต่อกับฐานข้อมูล MySQL ได้
2. เพื่อให้นักศึกษาสามารถพัฒนาโปรแกรมในการ Update และ Delete ข้อมูลในฐานข้อมูล MySQL โดยใช้ NodeJS และ Express

หน้าจอการทำงาน

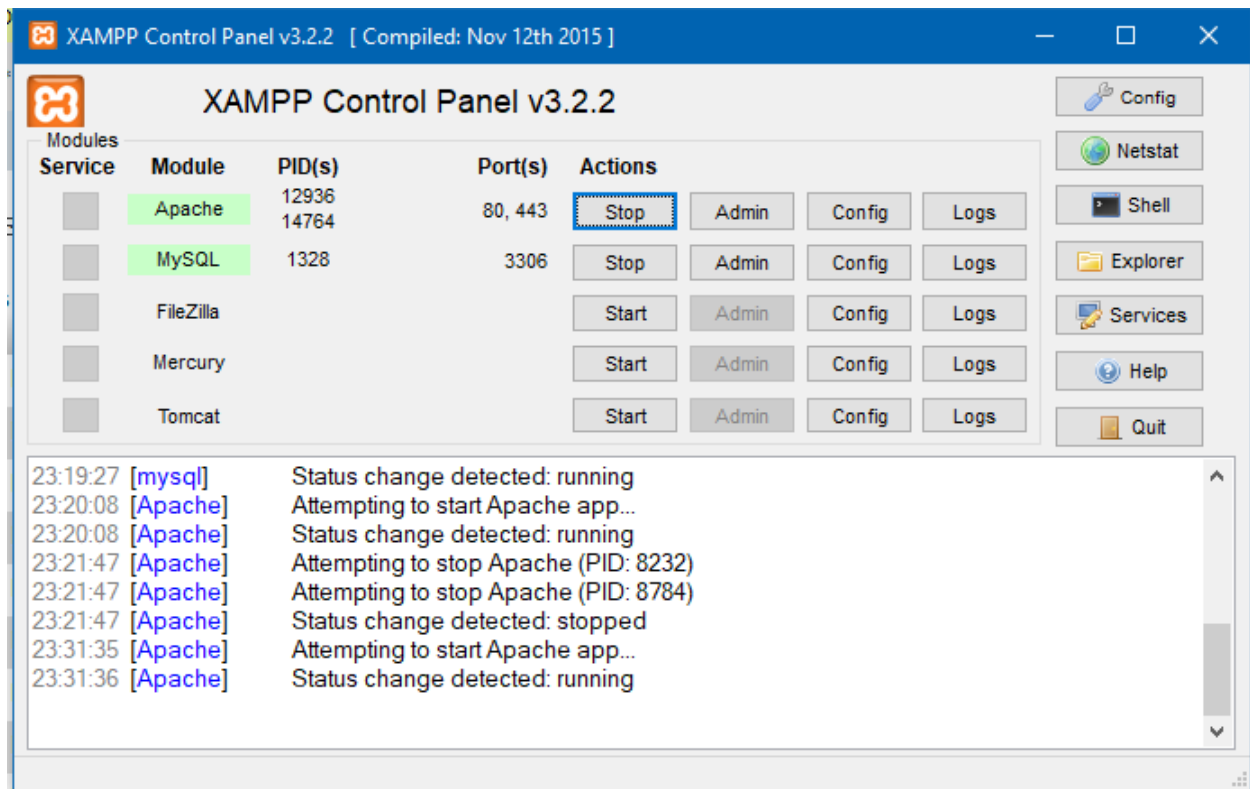
การทำงานหน้าจอแรกจะเป็นการดึงข้อมูลทั้งหมดจากฐานข้อมูลมาแสดง ในหน้าแรกยังมีส่วนที่สามารถค้นหาข้อมูล แก้ไขและลบข้อมูลนักศึกษาในฐานข้อมูลได้ แล้วแสดงข้อมูลที่เปลี่ยนแปลงหน้าจอแรก ดังภาพด้านล่าง





การใช้ฐานข้อมูลที่ MySQL

ในขั้นตอนแรกให้เปิดโปรแกรม XAMPP ให้ Start : Apache และ MySQL



จากนั้นให้เข้าที่ phpMyAdmin เพื่อเปิดฐานข้อมูลที่เคยสร้างไว้ใน Lab ที่ผ่านมาชื่อว่า lab_connect_mysql ซึ่งมีโครงสร้างและข้อมูล ดังนี้

Name	Type	Collation	Attributes	Null	Default	Comments	Extra
No	int(11)			No	None		AUTO_INCREMENT
std_id	varchar(12)	utf8_general_ci		No	None		
std_name	varchar(50)	utf8_general_ci		No	None		
std_age	int(2)			No	None		

No	std_id	std_name	std_age
1	63000000-1	Tony	20
2	63000002-2	Green	22
3	63000033-3	Sam	22
4	63000000-4	Alice	20
5	63000005-5	Dave	21

การสร้างไฟล์ API

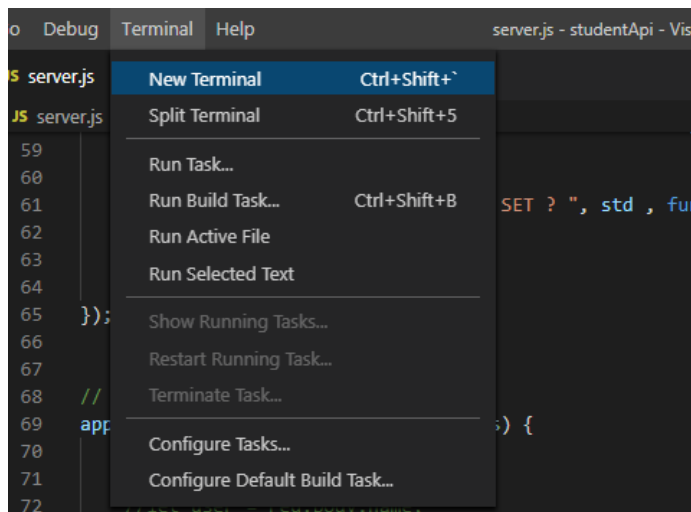
1. จากนั้นเปิดโปรแกรม VS Code แล้วเปิดไฟล์ server.js ไฟล์เดิมจาก Lab ที่แล้ว และให้เพิ่มคำสั่งในส่วนของการค้นหานักศึกษาจากรหัสนักศึกษา การUpdate และการ Delete โดยเขียนคำสั่งก่อน // set port

```
52 // Retrieve student with std_id
53 app.get('/std/:id', function (req, res) {
54
55     let std_id = req.params.id;
56
57     if (!std_id) {
58         return res.status(400).send({ error: true, message: 'Please provide std_id' });
59     }
60
61     dbConn.query('SELECT * FROM student where std_id=?', std_id, function (error, results, fields) {
62         if (error) throw error;
63         return res.send(results[0]);
64     });
65
66 });
67
```

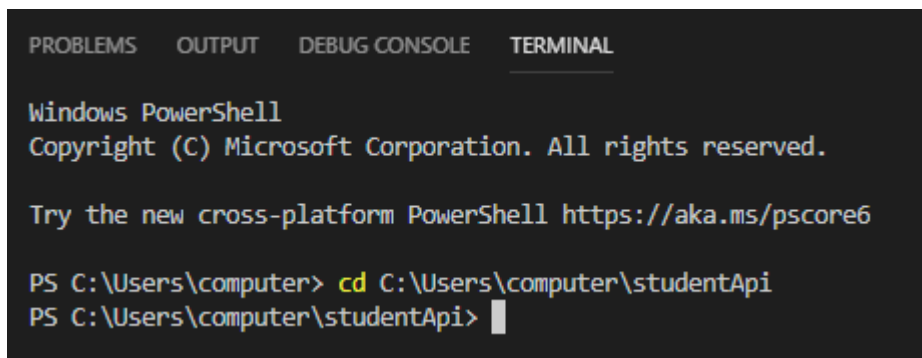
```
68 // Update student with id
69 app.put('/std/:id', function (req, res) {
70
71     let std_id = req.params.id;
72     let std = req.body;
73
74     if (!std_id || !std) {
75         return res.status(400).send({ error: user, message: 'Please provide student data and student_id' });
76     }
77
78     dbConn.query("UPDATE student SET ? WHERE std_id = ?", [std, std_id], function (error, results, fields) {
79         if (error) throw error;
80         return res.send({ error: false, data: results, message: 'Student has been updated successfully.' });
81     });
82 });
83
84 // Delete student
85 app.delete('/std/:id', function (req, res) {
86
87     let std_id = req.params.id;
88
89     if (!std_id) {
90         return res.status(400).send({ error: true, message: 'Please provide std_id' });
91     }
92     dbConn.query('DELETE FROM student WHERE std_id = ?', std_id, function (error, results, fields) {
93         if (error) throw error;
94         return res.send({ error: false, data: results, message: 'Student has been deleted successfully.' });
95     });
96 });
97
```

```
98 // set port
99 app.listen(3000, function () {
100     console.log('Node app is running on port 3000');
101 });
102
103 module.exports = app;
```

2. จากนั้นให้เปิด Terminal

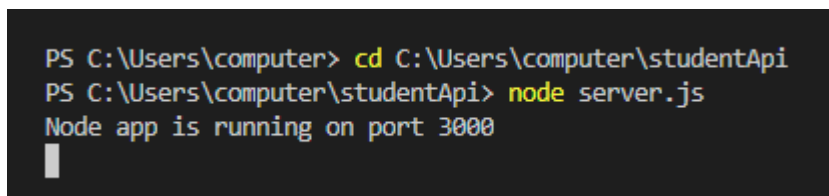


3. ถัดมาพิมพ์คำสั่งเพื่อเข้าไปที่โฟลเดอร์ที่ได้สร้างไว้ เช่น `cd C:\Users\computer\studentApi`



4. ให้ Run Server เพื่อเปิด port รอให้บริการ

`node server.js`



หรือ `nodemon server.js`

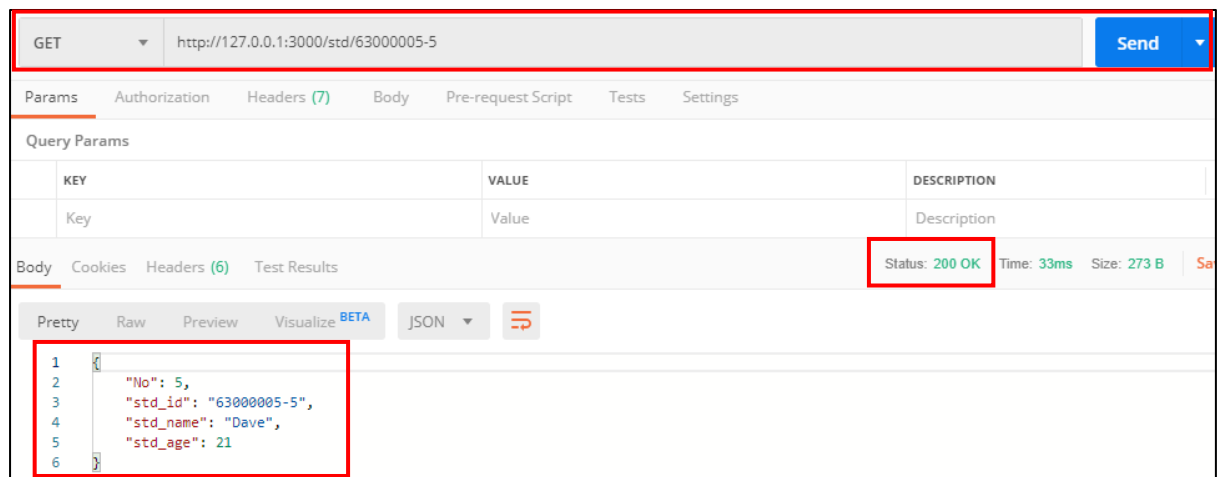
```
PS C:\Users\computer\studentApi> nodemon server.js
[nodemon] 1.19.2
[nodemon] to restart at any time, enter `rs`
[nodemon] watching dir(s): *.*
[nodemon] starting `node server.js`
Node app is running on port 3000
```

การใช้ postman ในการทดสอบ API

การค้นหาข้อมูลของนักศึกษาตามรหัสนักศึกษา

ให้กำหนดวิธีส่งข้อมูลเป็น GET และใส่ URL เป็น <http://127.0.0.1:3000/std/63000005-5> จากนั้นกดปุ่ม

Send

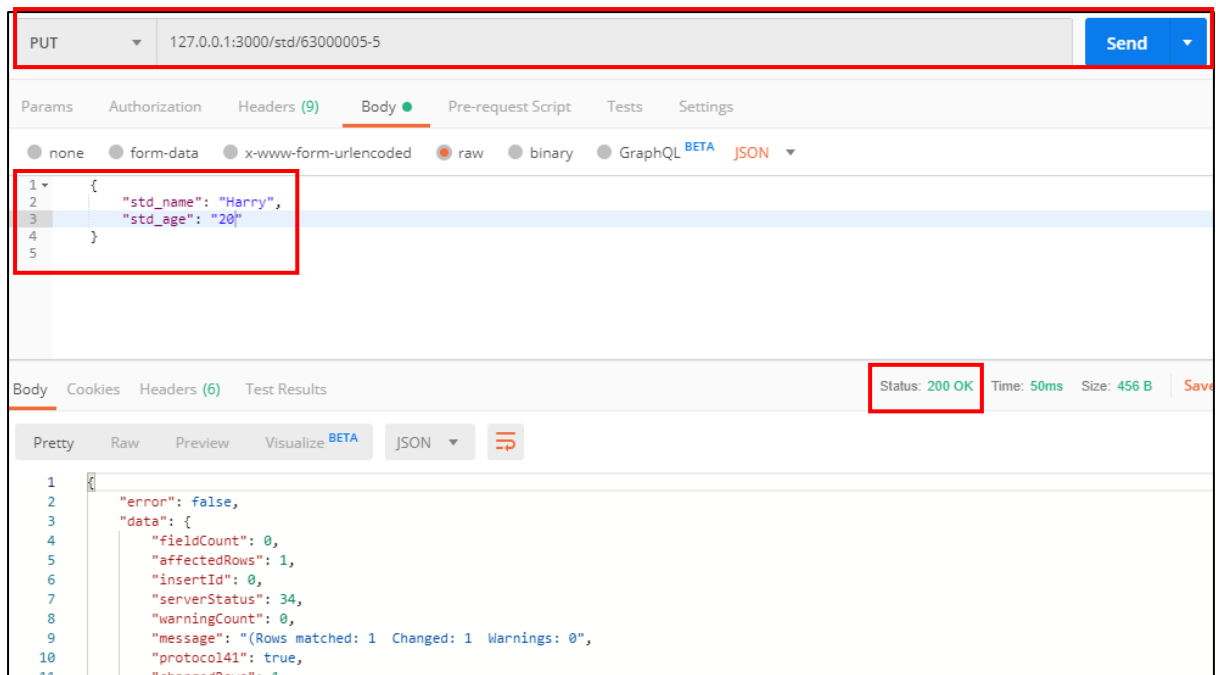


การแก้ไขข้อมูลของนักศึกษา (Update)

ให้กำหนดวิธีส่งข้อมูลเป็น PUT และใส่ URL เป็น <http://127.0.0.1:3000/std/63000005-5> และในส่วนของ body จะมีการกำหนดค่าเพื่อส่งข้อมูลในการแก้ไขข้อมูลในตาราง student ดังนี้

```
{
  "std_name": "Harry",
  "std_age": "20"
}
```

และจะต้องกำหนด Content-type เป็น JSON(application/json) จากนั้นกดปุ่ม Send



ผลลัพธ์ เมื่อเข้าไปที่ phpMyAdmin จะเห็นข้อมูลมีการเปลี่ยนแปลง

No	std_id	std_name	std_age
1	63000000-1	Tony	20
2	63000002-2	Green	22
3	63000033-3	Sam	22
4	63000000-4	Alice	20
5	63000005-5	Harry	20

การลบข้อมูลนักศึกษาที่กำหนด

ให้กำหนดวิธีส่งข้อมูลเป็น DELETE และใส่ URL เป็น <http://127.0.0.1:3000/std/63000005-5> จากนั้นกดปุ่ม Send

DELETE

http://127.0.0.1:3000/std/63000005-5

Send

ParamsAuthorizationHeaders (8)BodyPre-request ScriptTestsSettings

Query Params

KEY	VALUE	DESCRIPTION
Key	Value	Description

BodyCookiesHeaders (6)Test Results

Status: 200 OKTime: 31msSize: 415 BSave

PrettyRawPreviewVisualize BETAJSON

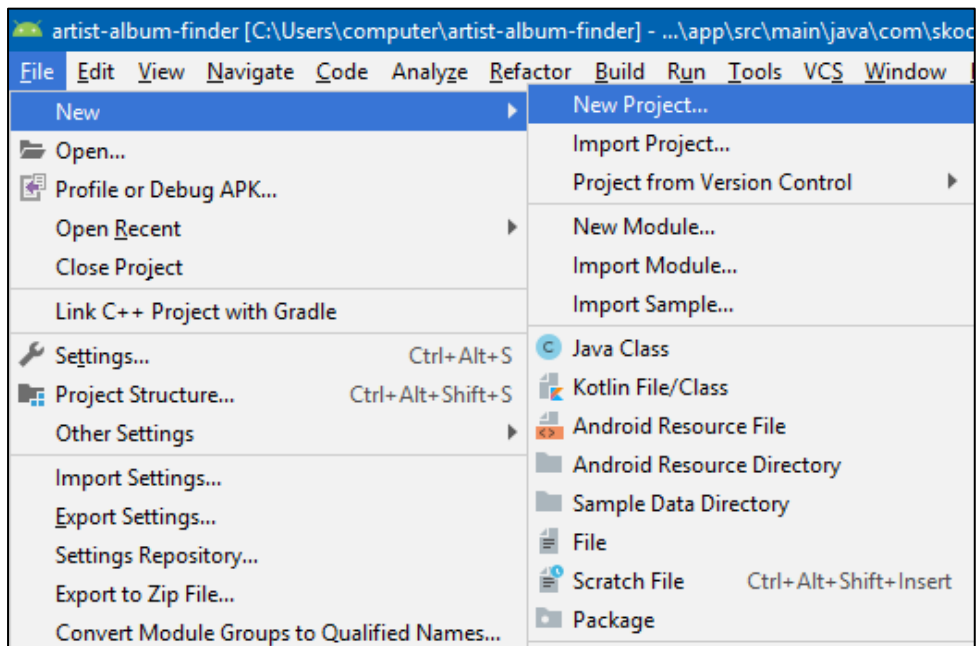
```
1  {
2    "error": false,
3    "data": {
4      "fieldCount": 0,
5      "affectedRows": 1,
6      "insertId": 0,
7      "serverStatus": 34,
8      "warningCount": 0,
9      "message": "",
10     "protocol41": true,
11     "changedRows": 0
12   },
13   "message": "Student has been deleted successfully."
14 }
```

ผลลัพธ์ เมื่อเข้าไปที่ phpMyAdmin จะเห็นข้อมูลถูกลบไป

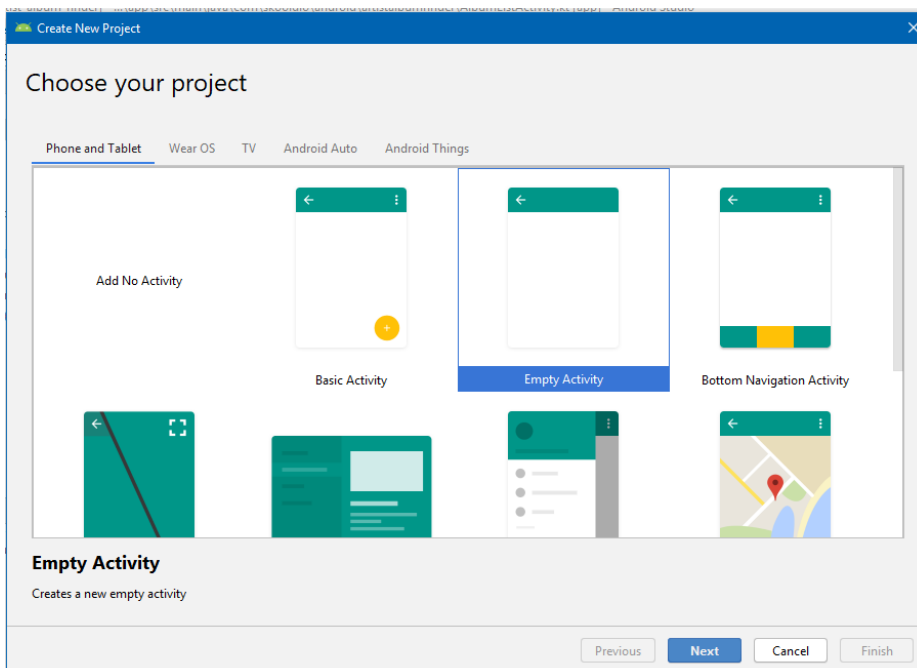
No	std_id	std_name	std_age
1	63000000-1	Tony	20
2	63000002-2	Green	22
3	63000033-3	Sam	22
4	63000000-4	Alice	20

การเขียนคำสั่งในส่วนของ Android

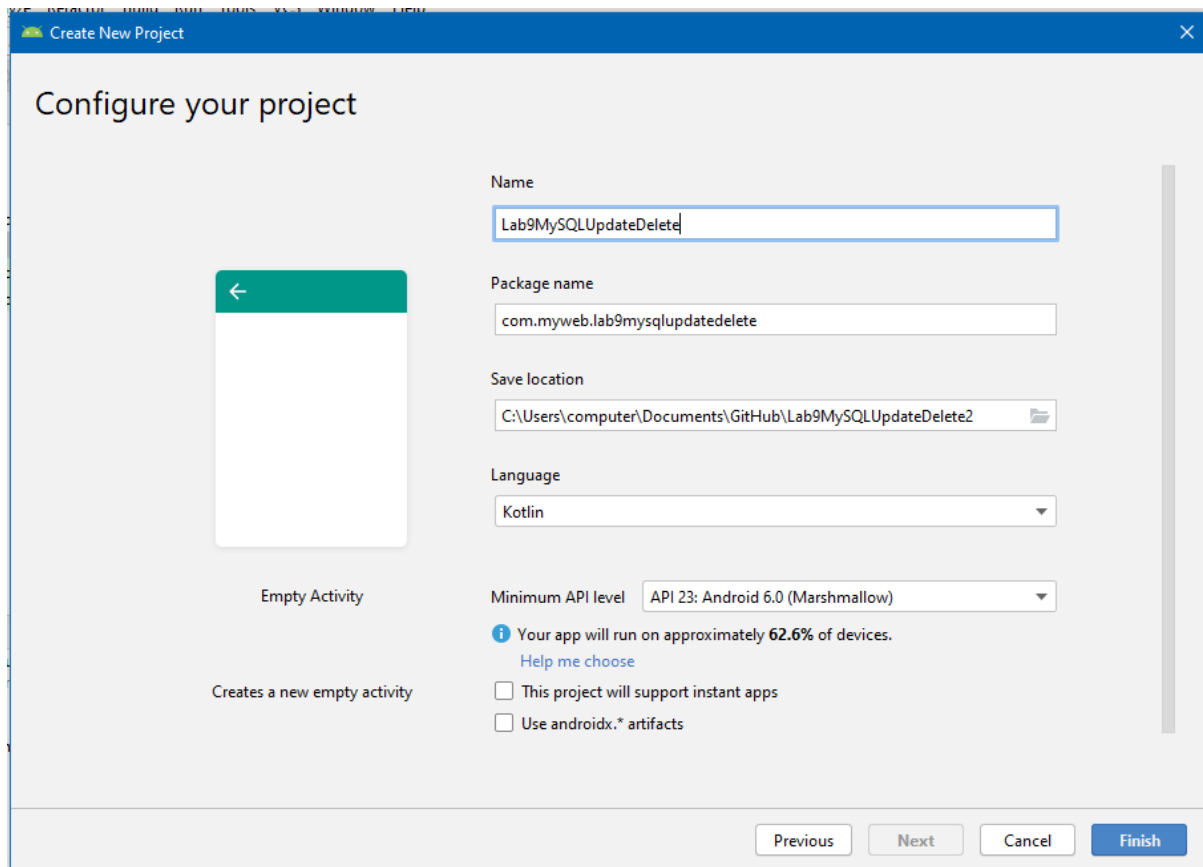
1. เมื่อเปิดโปรแกรม Android Studio แล้วไปที่เมนู File แล้วเลือก New คลิกที่ New Project



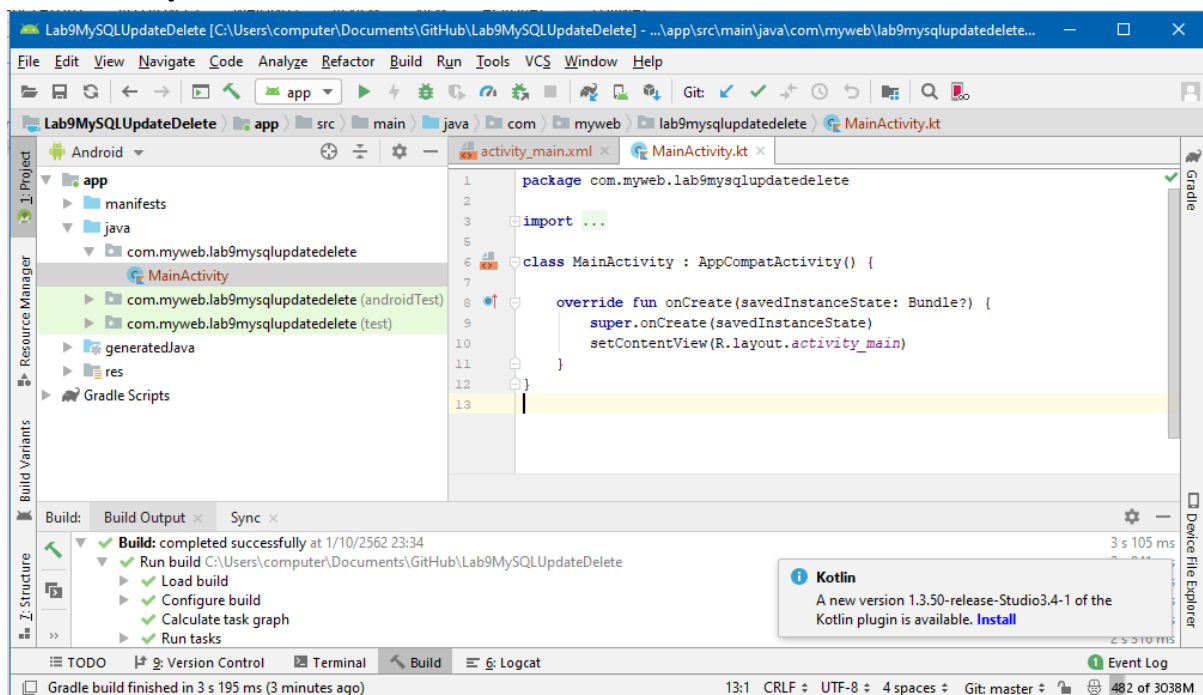
2. จากนั้นจะปรากฏหน้าจอ ดังนี้ ให้เลือก Empty Activity จากนั้นกด Next



3. ถัดมาให้ตั้งชื่อ Application name คือ Lab9MySQLUpdateDelete และ Language เป็น Kotlin จากนั้นกด Finish

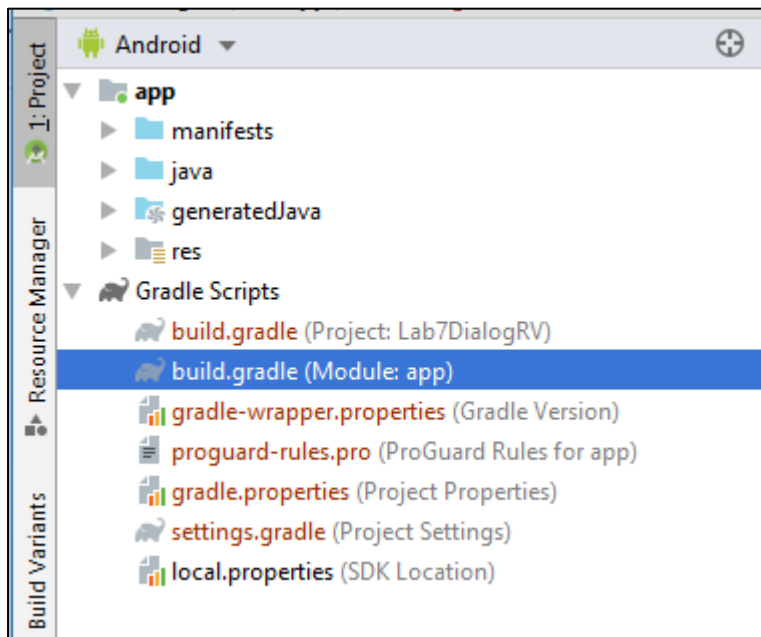


จะต้องรอสักครู່ เพื่อให้โปรแกรมสร้างโปรเจกเสร็จ



การเพิ่ม Library ของไฟล์ app.Gradle

1. ให้ไปที่ Gradle Scripts คลิกที่ build.gradle(Module:app)



2. ให้เพิ่มคำสั่งดังนี้

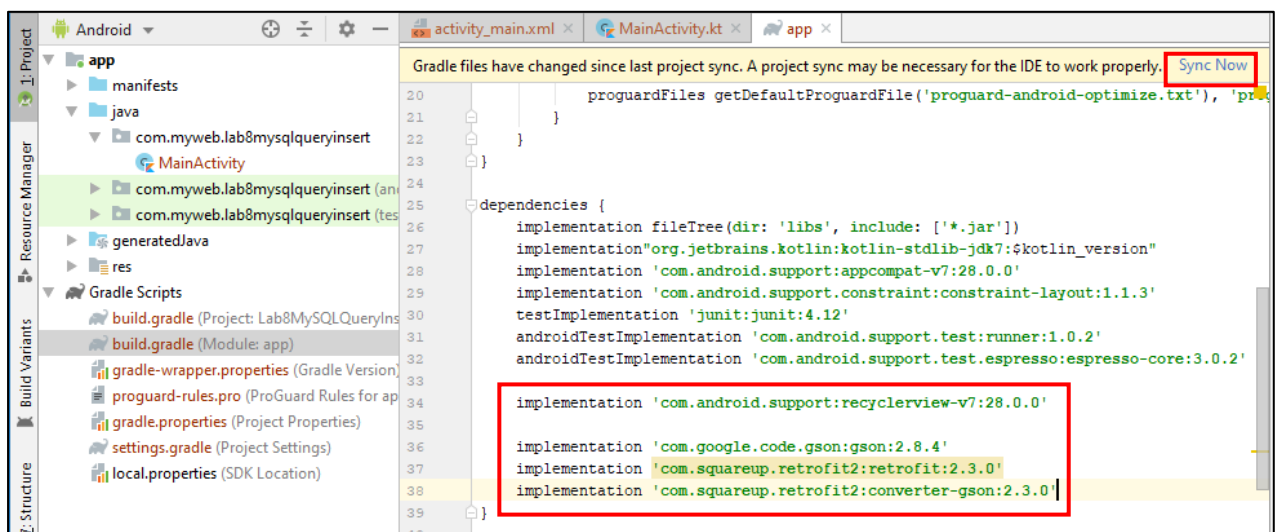
```
implementation 'com.google.code.gson:gson:2.8.4'
```

```
implementation 'com.squareup.retrofit2:retrofit:2.3.0'
```

```
implementation 'com.squareup.retrofit2:converter-gson:2.3.0'
```

```
implementation 'com.android.support:recyclerview-v7:28.0.0'
```

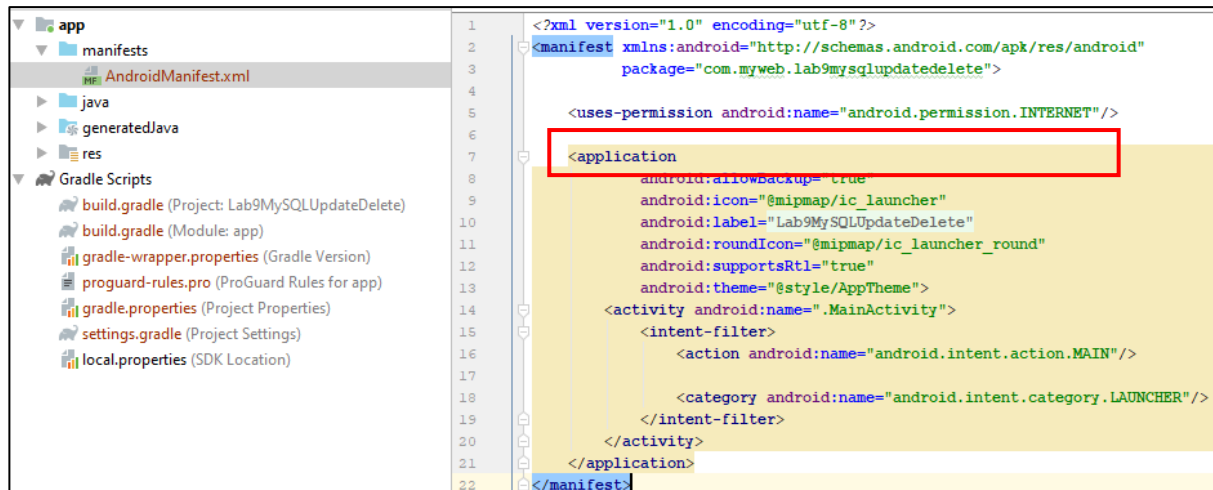
จากนั้นให้คลิกที่ Sync Now ด้านบนขวา เพื่อโหลด Library ของ RecyclerView, Retrofit และ Gson



ส่วนของไฟล์ AndroidManifest.xml

ให้เข้าไปที่ app >> manifests >> AndroidManifest.xml จากนั้นเพิ่มคำสั่ง

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



โครงสร้างของโปรเจค

การออกแบบ Layout

- activity_main.xml >> หน้า main
- edit_delete_item_layout.xml >> การออกแบบส่วน item ของ RecyclerView
- activity_edit_students.xml >> หน้าจอแก้ไขและลบข้อมูล

การสร้างการทำงาน

- Student.kt >> Data Class ของนักศึกษา
- EditStudentsAdapter.kt >> Adapter ที่ทำงานกับ RecyclerView
- StudentAPI.kt >> interface ทำงานกับ API
- EditStudentsActivity.kt >> การทำงานของแก้ไขและลบข้อมูล
- MainActivity.kt >> Class สำหรับจัดการเกี่ยวกับ UI

การออกแบบ Layout

การเพิ่มคำสั่ง ไฟล์ activity_main.xml ดังนี้

```

1  <?xml version="1.0" encoding="utf-8"?>
2  <LinearLayout
3      xmlns:android="http://schemas.android.com/apk/res/android"
4      xmlns:tools="http://schemas.android.com/tools"
5      android:layout_width="match_parent"
6      android:layout_height="match_parent"
7      android:orientation="vertical"
8      tools:context=".MainActivity">
9      <LinearLayout
10         android:layout_width="match_parent"
11         android:layout_height="wrap_content"
12         android:padding="10dp"
13         android:gravity="center"
14         android:orientation="horizontal">
15         <EditText
16             android:id="@+id/edt_search"
17             android:layout_width="200dp"
18             android:layout_height="wrap_content"
19             android:hint="Student ID"/>
20         <Button
21             android:layout_width="wrap_content"
22             android:layout_height="wrap_content"
23             android:text="Search"
24             android:onClick="clickSearch"/>
25     </LinearLayout>
26
27     <TextView
28         android:id="@+id/text1"
29         android:layout_width="match_parent"
30         android:layout_height="wrap_content"
31         android:text="Edit/Delete Student"
32         android:textAlignment="center"
33         android:layout_marginBottom="20dp"
34         android:textSize="20sp" />
35     <android.support.v7.widget.RecyclerView
36         android:id="@+id/recycler_view"
37         android:paddingBottom="100dp"
38         android:clipToPadding="false"
39         android:layout_width="match_parent"
40         android:layout_height="match_parent">
41     </android.support.v7.widget.RecyclerView>
42 </LinearLayout>

```

ส่วนของ design จะแสดงดังนี้

Student ID

SEARCH

Edit/Delete Student

Item 0

Item 1

Item 2

Item 3

Item 4

Item 5

Item 6

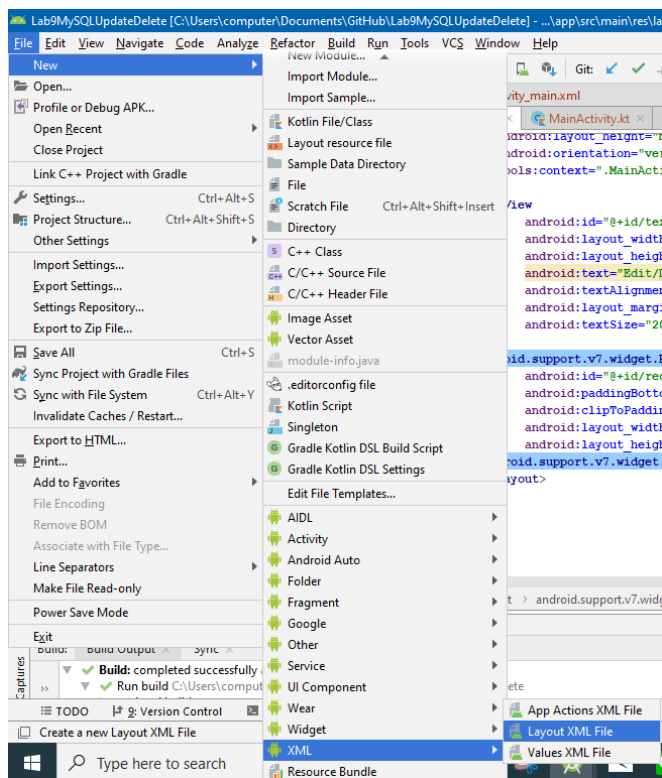
Item 7

Item 8

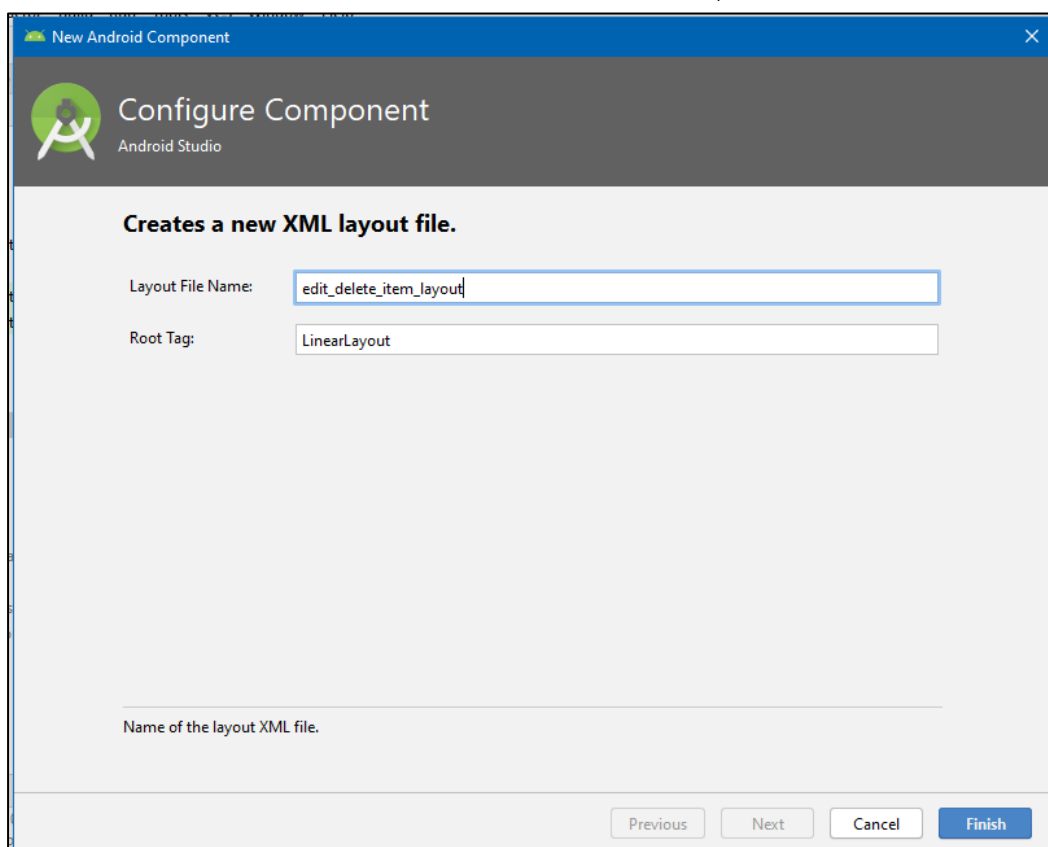
Item 9

การเพิ่มคำสั่งในไฟล์ edit_delete_item_layout.xml ดังนี้

1. ให้ไปที่ File >> New >> XML >> Layout XML File



2. จากนั้นให้พิมพ์ชื่อ layout คือ edit_delete_item_layout แล้วคลิกปุ่ม Finish



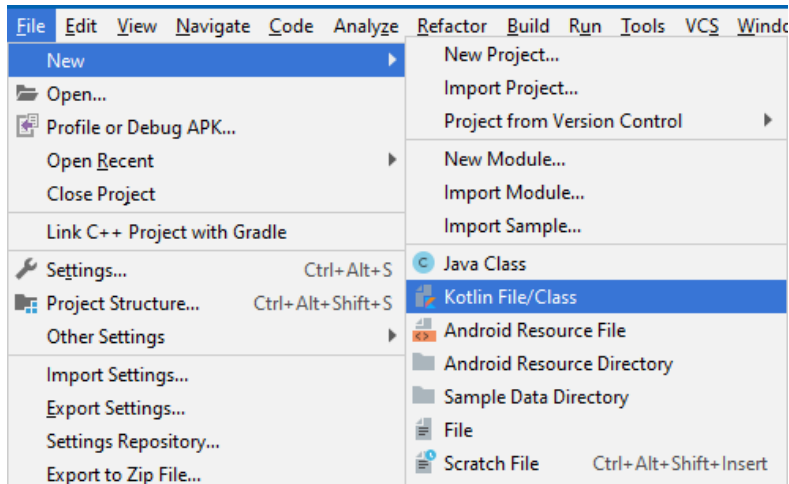
หลังจากนั้นให้เพิ่มคำสั่ง ดังภาพด้านล่าง

```
1  <?xml version="1.0" encoding="utf-8"?>
2  <RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
3      android:layout_width="match_parent"
4      android:layout_height="wrap_content">
5      <LinearLayout
6          android:layout_width="match_parent"
7          android:layout_height="wrap_content"
8          android:layout_toLeftOf="@+id/editBTN"
9          android:layout_toStartOf="@+id/editBTN"
10         android:gravity="center_vertical"
11         android:orientation="horizontal">
12         <TextView
13             android:id="@+id/tvID"
14             android:layout_width="wrap_content"
15             android:layout_height="wrap_content"
16             android:layout_marginLeft="18dp"
17             android:layout_marginRight="28dp"
18             android:text="ID"
19             android:textColor="#313131"
20             android:textSize="20sp"
21             android:textStyle="bold" />
22         <TextView
23             android:id="@+id/tvName"
24             android:layout_width="match_parent"
25             android:layout_height="wrap_content"
26             android:layout_marginLeft="18dp"
27             android:layout_marginRight="28dp"
28             android:text="Name"
29             android:textColor="#313131"
30             android:textSize="20sp"
31             android:textStyle="bold" />
32     </LinearLayout>
33     <TextView
34         android:id="@+id/editBTN"
35         android:layout_width="80dp"
36         android:layout_height="wrap_content"
37         android:layout_alignParentEnd="true"
38         android:layout_alignParentRight="true"
39         android:layout_centerVertical="true"
40         android:background="#F0F7F8F8"
41         android:gravity="center"
42         android:text="EDIT/DELETE"
43         android:textColor="#fc6b10"
44         android:textSize="15sp"
45         android:textStyle="bold" />
46 </RelativeLayout>
```

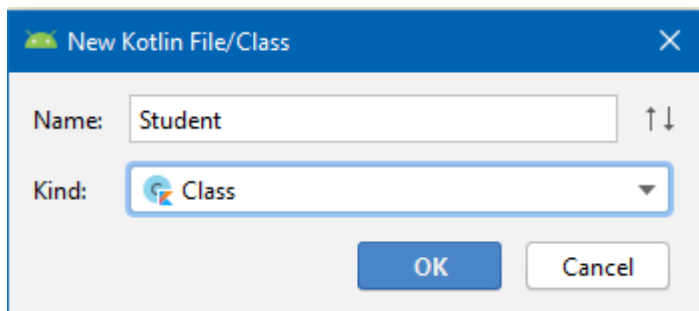
การสร้างการทำงานส่วน File Kotlin

สร้าง Data Class ของ Student.kt (ใช้คำสั่งเดิมจาก Lab 8 ได้)

1. ให้ไปที่ File >> New >> Kotlin File/Class



2. จากนั้นจะแสดงหน้าจอให้กรอกชื่อคลาสชื่อ Student และกำหนดชนิดไฟล์เป็น Class

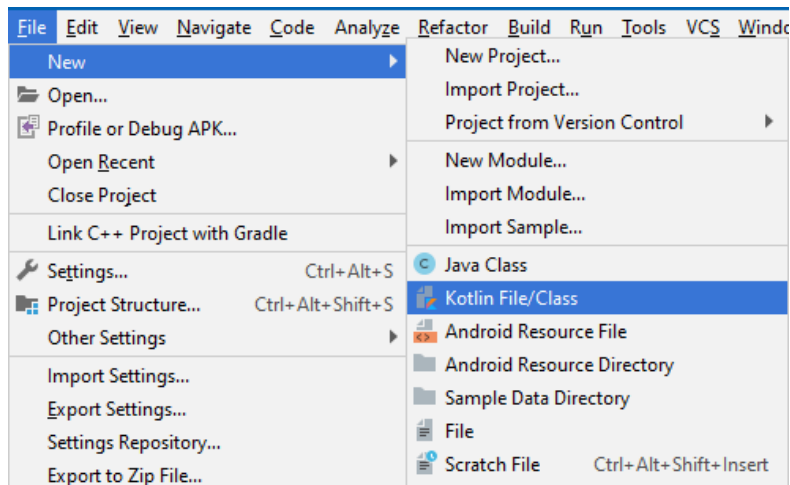


3. แล้วให้พิมพ์โค้ดคำสั่ง โดยกำหนดให้เป็น data class และให้มี รหัสนักศึกษา (id), ชื่อนักศึกษา (name) เป็นประเภท String และอายุ (age) เป็นประเภท Integer

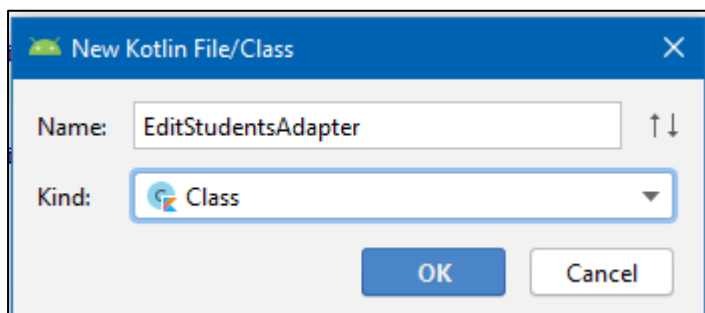
```
1 package com.myweb.lab8mysqlqueryinsert
2
3 import com.google.gson.annotations.Expose
4 import com.google.gson.annotations.SerializedName
5
6 data class Student(
7     @Expose
8     @SerializedName( value: "std_id") val std_id: String,
9
10    @Expose
11    @SerializedName( value: "std_name") val std_name: String,
12
13    @Expose
14    @SerializedName( value: "std_age") val std_age: Int) {}
```

การสร้างไฟล์ EditStudentsAdapter.kt

1. ให้ไปที่ File >> New >> Kotlin File/Class

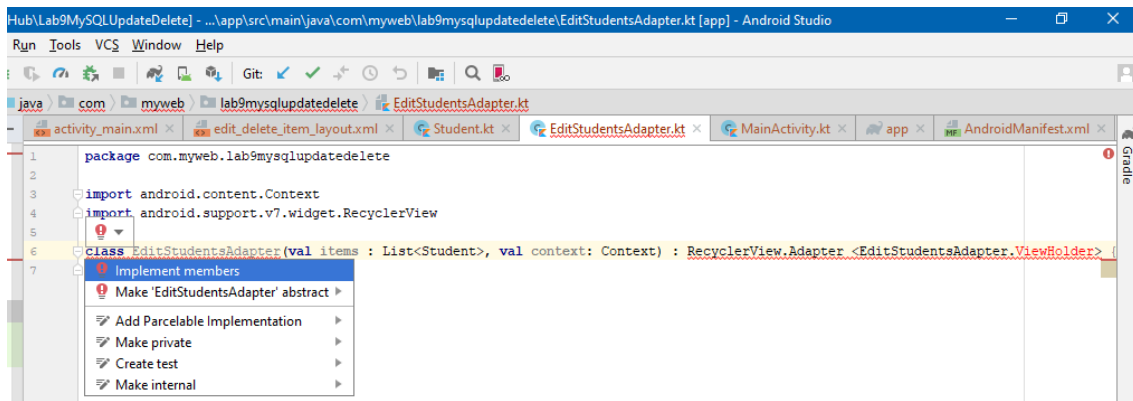


2. จากนั้นจะแสดงหน้าจอให้กรอกชื่อคลาสชื่อ EditStudentsAdapter

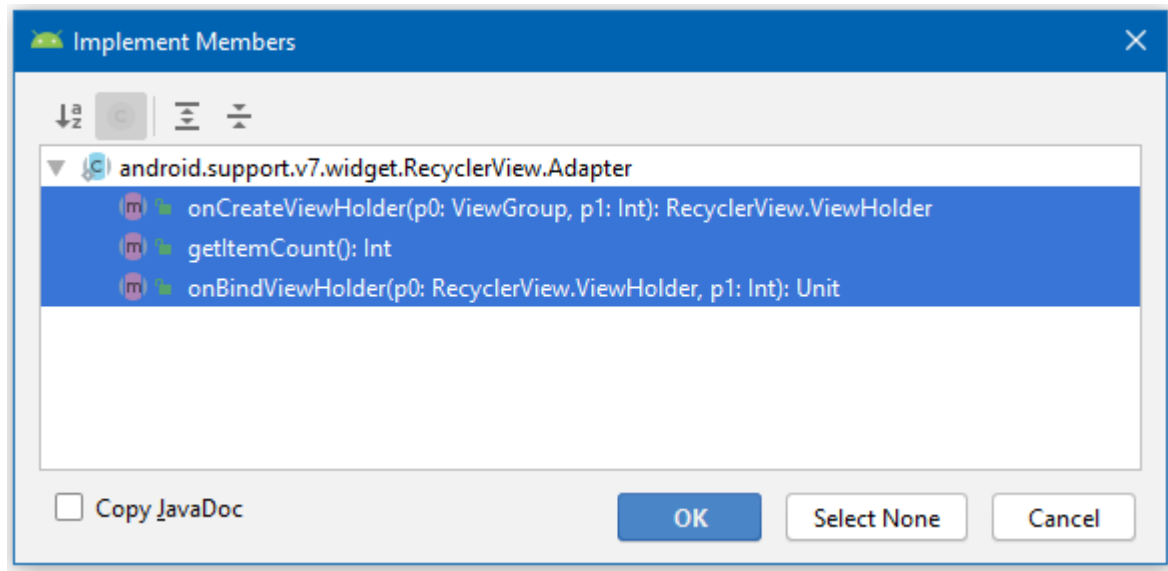


3. สำหรับ Class **EditStudentsAdapter** ทำการ extends RecyclerView.Adapter < EditStudentsAdapter. EditViewHolder> โดยสร้างตัวแปรประเภท List<Student> และ Context แล้วกำหนดค่าผ่าน constructor

เมื่อแสดงข้อความ Error ให้คลิกที่  แล้วเลือก Implement members



จากนั้นจะขึ้นหน้าต่างให้เลือกทุก Function แล้วคลิกปุ่ม OK



แสดงหน้าจอดังนี้

```

1 package com.myweb.lab9mysqlupdatedelete
2
3 import android.content.Context
4 import android.support.v7.widget.RecyclerView
5 import android.view.ViewGroup
6
7 class EditStudentsAdapter(val items : List<Student>, val context: Context) : RecyclerView.Adapter<EditStudentsAdapter.ViewHolder> {
8     override fun onCreateViewHolder(p0: ViewGroup, p1: Int): EditStudentsAdapter.ViewHolder {
9         TODO( reason: "not implemented") //To change body of created functions use File | Settings | File Templates.
10    }
11
12    override fun getItemCount(): Int {
13        TODO( reason: "not implemented") //To change body of created functions use File | Settings | File Templates.
14    }
15
16    override fun onBindViewHolder(p0: EditStudentsAdapter.ViewHolder, p1: Int) {
17        TODO( reason: "not implemented") //To change body of created functions use File | Settings | File Templates.
18    }
19 }

```

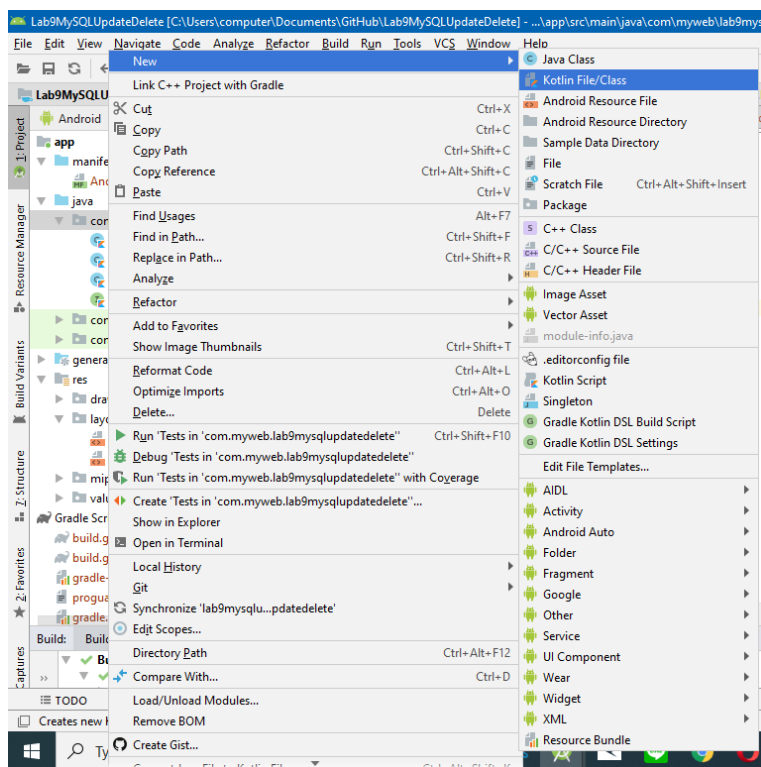
หลังจากนั้นให้เพิ่มคำสั่งสร้าง class “ViewHolder” extends RecyclerView.ViewHolder ในส่วนด้านล่างสุด

และให้แก้ไขคำสั่งด้านบนของการส่งค่ากลับของ Class “StudentsAdapter” และส่วนของฟังก์ชัน onCreateViewHolder, getItemCount, onBindViewHolder ใน Class ของ StudentsAdapter ดังนี้

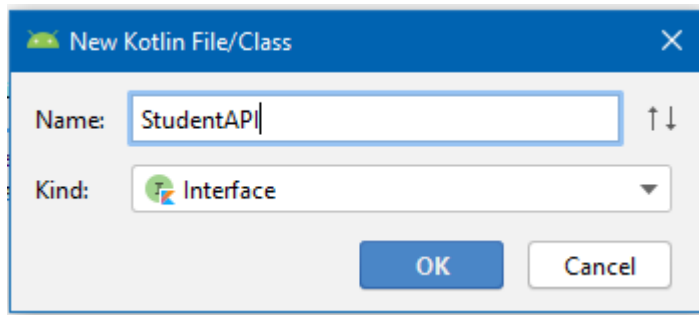
```
10 class EditStudentsAdapter(val items : List<Student>, val context: Context) :
11     RecyclerView.Adapter<EditStudentsAdapter.ViewHolder>() {
12
13     override fun onCreateViewHolder(parent: ViewGroup, viewType: Int): ViewHolder {
14         val view_item :View? = LayoutInflater.from(parent.context).inflate(R.layout.edit_delete_item_layout, parent, attachToRoot: false)
15         return ViewHolder(view_item)
16     }
17
18     override fun getItemCount(): Int {
19         return items.size
20     }
21
22     override fun onBindViewHolder(holder: ViewHolder, position: Int) {
23         holder.tvID?.text = items[position].std_id
24         holder.tvName?.text = items[position].std_name
25     }
26
27     class ViewHolder (view: View) : RecyclerView.ViewHolder(view) {
28         // Holds the TextView that will add each student to
29         val tvID :TextView? = view.tvID
30         val tvName :TextView? = view.tvName
31     }
32 }
```

การสร้างไฟล์ StudentAPI.kt สำหรับ interface ทำงานกับ API

1. ให้ไปที่ Flie >> New >> Kotlin File/Class



2. จากนั้นจะแสดงหน้าจอให้กรอกชื่อคลาสชื่อ StudentAPI และ Kind เป็น Interface แล้วคลิกที่ปุ่ม OK



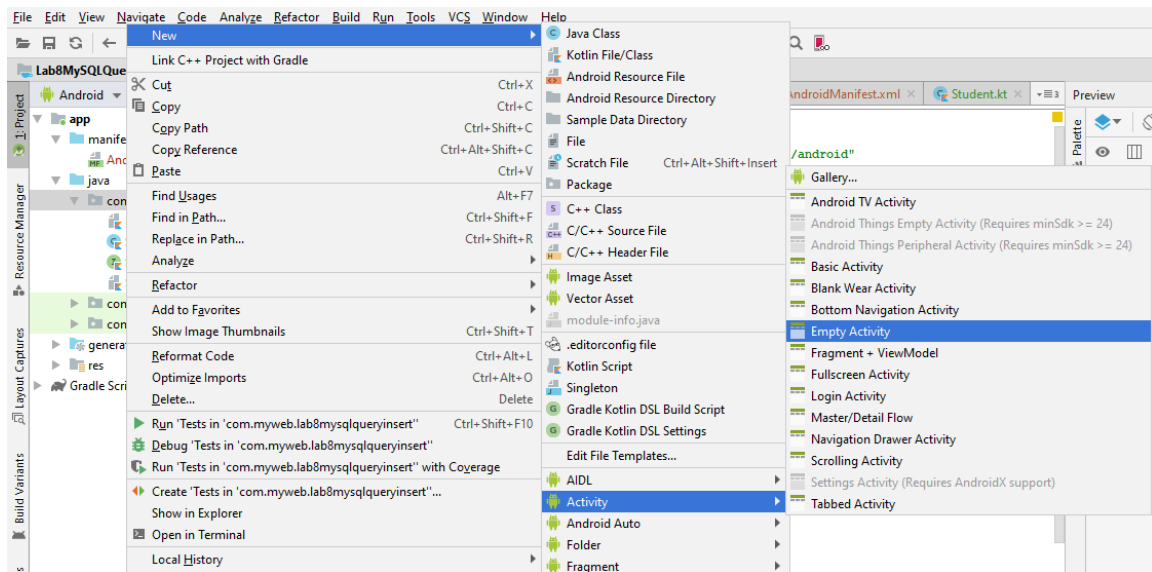
3. สำหรับ Interface ชื่อ StudentAPI ให้เพิ่มคำสั่ง ดังนี้

```
8 interface StudentAPI {
9     @GET( value: "allstd")
10     fun retrieveStudent(): Call<List<Student>>
11
12     @GET( value: "std/{std_id}")
13     fun retrieveStudentID(
14         @Path( value: "std_id") std_id: String): Call<Student>
15
16     @FormUrlEncoded
17     @PUT( value: "std/{std_id}")
18     fun updateStudent(
19         @Path( value: "std_id") std_id: String,
20         @Field( value: "std_name") std_name: String,
21         @Field( value: "std_age") std_age: Int): Call<Student>
22
23     @DELETE( value: "std/{std_id}")
24     fun deleteStudent(
25         @Path( value: "std_id") std_id: String): Call<Student>
26
27     companion object {
28         fun create(): StudentAPI {
29             val stdClient : StudentAPI = Retrofit.Builder()
30                 .baseUrl( baseUrl: "http://10.0.2.2:3000/")
31                 .addConverterFactory(GsonConverterFactory.create())
32                 .build()
33                 .create(StudentAPI ::class.java)
34             return stdClient
35         }
36     }
37 }
```

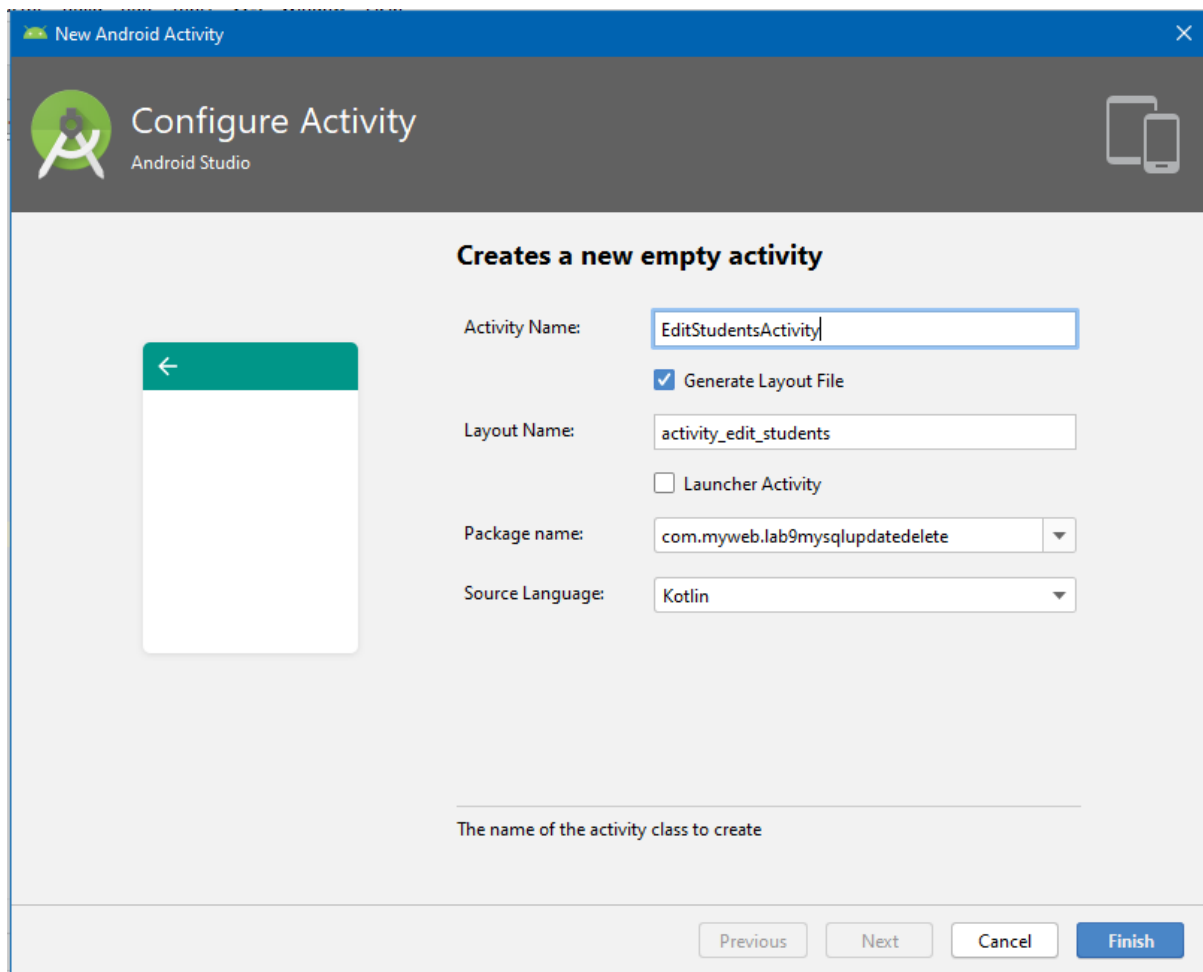
ในส่วนของ companion object ทำให้สามารถเรียกใช้ method ได้ใน Interface ได้

การสร้างหน้าจอในการ Update และ Delete

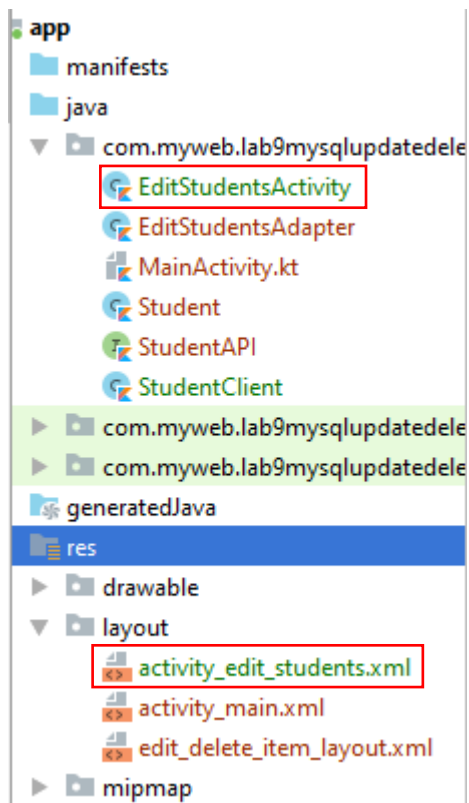
1. ให้สร้าง Activity ใหม่ขึ้นมา ชื่อ EditStudentsActivity ให้ไปที่ File >> New >> Activity >> เลือก Empty Activity



2. จากนั้นให้กรอก Activity Name คือ I EditStudentsActivity และต้องเลือก Source Language เป็น Kotlin แล้วกดปุ่ม Finish



3. จากนั้นโปรแกรมจะสร้างไฟล์ให้ 2 ไฟล์ คือ EditStudentsActivity.kt และ activity_edit_students.xml



การสร้างหน้าจอของ activity_edit_students.xml

1. หน้าจอ activity_edit_students จะเป็นการรับค่าจากผู้ใช้ เพื่อนำข้อมูลไปแก้ไขในตาราง Student ตาม std_id ที่กำหนด โดยพิมพ์คำสั่งดังนี้


```

1  <?xml version="1.0" encoding="utf-8"?>
2  <LinearLayout
3      xmlns:android="http://schemas.android.com/apk/res/android"
4      xmlns:tools="http://schemas.android.com/tools"
5      android:layout_width="match_parent"
6      android:layout_height="match_parent"
7      android:orientation="vertical"
8      tools:context=".EditStudentsActivity">
9      <TextView
10         android:layout_width="match_parent"
11         android:layout_height="wrap_content"
12         android:text="Edit OR Delete Student"
13         android:textStyle="bold"
14         android:textAlignment="center"
15         android:textSize="25sp"/>
16      <EditText
17         android:id="@+id/edit_id"
18         android:layout_width="match_parent"
19         android:layout_height="wrap_content"
20         android:inputType="text"
21         android:textSize="20sp"
22         android:hint="Input ID"/>
23      <EditText
24         android:id="@+id/edit_name"
25         android:layout_width="match_parent"
26         android:layout_height="wrap_content"
27         android:inputType="text"
28         android:textSize="20sp"
29         android:hint="Input Name"/>
30      <EditText
31         android:id="@+id/edit_age"
32         android:layout_width="match_parent"
33         android:layout_height="wrap_content"
34         android:inputType="number"
35         android:textSize="20sp"
36         android:hint="Input Age"/>
37      <Button
38         android:id="@+id/btnUpdate"
39         android:layout_width="match_parent"
40         android:layout_height="wrap_content"
41         android:layout_marginTop="10dp"
42         android:background="#00E676"
43         android:text="Seve"
44         android:textSize="20sp"
45         android:onClick="saveStudent" />

```

```

46 <Button
47     android:id="@+id/btnDelete"
48     android:layout_width="match_parent"
49     android:layout_height="wrap_content"
50     android:layout_marginTop="10dp"
51     android:background="#FF1744"
52     android:text="Delete"
53     android:textSize="20sp"
54     android:onClick="deleteStudent" />
55 </LinearLayout>

```

ส่วนของ design จะแสดงดังนี้

2. ถัดมาพิมพ์คำสั่งในไฟล์ EditStudentsActivity ดังนี้

```

16 class EditStudentsActivity : AppCompatActivity() {
17     val createClient : StudentAPI = StudentAPI.create()
18     override fun onCreate(savedInstanceState: Bundle?) {
19         super.onCreate(savedInstanceState)
20         setContentView(R.layout.activity_edit_students)
21
22         val mId : String! = intent.getStringExtra( name: "mId")
23         val mName : String! = intent.getStringExtra( name: "mName")
24         val mAge : String! = intent.getStringExtra( name: "mAge")
25
26         edit_id.setText(mId)
27         edit_id.isEnabled=false
28         edit_name.setText(mName)
29         edit_age.setText(mAge)
30     }
31
32     fun saveStudent(v: View){
33         createClient.updateStudent(
34             edit_id.text.toString(),
35             edit_name.text.toString(),
36             edit_age.text.toString().toInt()).enqueue(object : Callback<Student> {

```

```

37  override fun onResponse(call: Call<Student>, response: Response<Student>) {
38      if (response.isSuccessful) {
39          Toast.makeText(applicationContext, text: "Successfully Updated", Toast.LENGTH_SHORT).show()
40          finish()
41      }else{
42          Toast.makeText(applicationContext, text: "Error ", Toast.LENGTH_SHORT).show()
43      }
44  }
45  override fun onFailure(call: Call<Student>, t: Throwable) {
46      Toast.makeText(applicationContext, text: "Error onFailure " + t.message, Toast.LENGTH_LONG).show()
47  }
48  })
49  }
50
51  fun deleteStudent(v: View){
52      /// Add instruction to Delete Student
53  }
54  }

```

การเพิ่มคำสั่งในไฟล์ MainActivity.kt

```

19 class MainActivity : AppCompatActivity() {
20     var studentList : ArrayList<Student> = arrayListOf<Student>()
21     val createClient : StudentAPI = StudentAPI.create()
22     override fun onCreate(savedInstanceState: Bundle?) {
23         super.onCreate(savedInstanceState)
24         setContentView(R.layout.activity_main)
25
26         recycler_view.layoutManager = LinearLayoutManager(applicationContext) as RecyclerView.LayoutManager?
27         recycler_view.itemAnimator = DefaultItemAnimator() as RecyclerView.ItemAnimator?
28         recycler_view.addItemDecoration(DividerItemDecoration(recycler_view.getContext(), DividerItemDecoration.VERTICAL))
29
30         recycler_view.addItemTouchListener(object : OnItemClickListener {
31             override fun onItemClick(position: Int, view: View) {
32                 Toast.makeText(applicationContext, text: "You click on : "+studentList[position].std_id,
33                     Toast.LENGTH_SHORT).show()
34                 val std : Student = studentList[position]
35                 val intent = Intent(applicationContext, EditStudentsActivity::class.java)
36                 intent.putExtra(name: "mId", std.std_id)
37                 intent.putExtra(name: "mName", std.std_name)
38                 intent.putExtra(name: "mAge", std.std_age.toString())
39                 startActivity(intent)
40             }
41         })
42     }
43
44     override fun onResume() {
45         super.onResume()
46         callStudentData()
47     }
48
49     fun clickSearch(v:View) {
50         studentList.clear()
51         createClient.retrieveStudentID(edt_search.text.toString())
52         .enqueue(object : Callback<Student> {
53
54             override fun onResponse(call: Call<Student>, response: Response<Student>) {
55                 studentList.add(Student(response.body()?.std_id.toString(),response.body()?.std_name.toString(),
56                     response.body()?.std_age.toString().toInt()))
57                 //// Set Data to RecyclerView
58                 recycler_view.adapter = EditStudentsAdapter(studentList,applicationContext)
59             }
60             override fun onFailure(call: Call<Student>, t: Throwable) : Unit = t.printStackTrace()
61         })
62     }
63
64     fun callStudentData(){
65         studentList.clear()
66         createClient.retrieveStudent()
67         .enqueue(object : Callback<List<Student>> {
68             override fun onResponse(call: Call<List<Student>>, response: Response<List<Student>>) {
69                 response.body()?.forEach { it: Student
70                     studentList.add(Student(it.std_id, it.std_name,it.std_age))
71                 }
72             }
73         })
74     }

```

```

72      /// Set Data to RecyclerView
73      recycler_view.adapter = EditStudentsAdapter(studentList,applicationContext)
74  }
75  override fun onFailure(call: Call<List<Student>>, t: Throwable) :Unit = t.printStackTrace()
76  })
77  }
78  }
79
80  interface OnItemClickListener {
81      fun onItemClick(position: Int, view: View)
82  }
83  fun RecyclerView.setOnItemClickListener(onClickListener: OnItemClickListener) {
84      this.addChildAttachStateChangeListener(object: RecyclerView.OnChildAttachStateChangeListener {
85
86          override fun onChildViewDetachedFromWindow(view: View) {
87              view?.setOnClickListener(null)
88          }
89
90          override fun onChildViewAttachedToWindow(view: View) {
91              view?.setOnClickListener { it: View!
92                  val holder : RecyclerView.ViewHolder! = getChildViewHolder(view)
93                  onClickListener.onItemClick(holder.adapterPosition, view)
94              }
95          }
96      })
97  }

```

Assignment 9

ให้นักศึกษาเขียนคำสั่งต่อจากโปรเจกต์ที่สร้างไว้ เพื่อให้สามารถลบข้อมูลของนักศึกษาได้ในส่วนของไฟล์ EditStudentsActivity.kt ให้สามารถทำงานได้สมบูรณ์

