

## ANDROID DEVELOPMENT

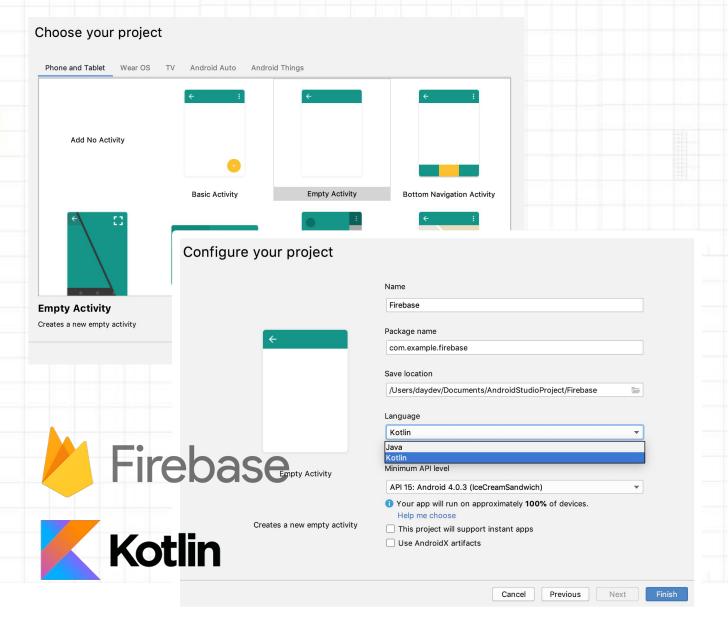
Banyapon Poolsawas



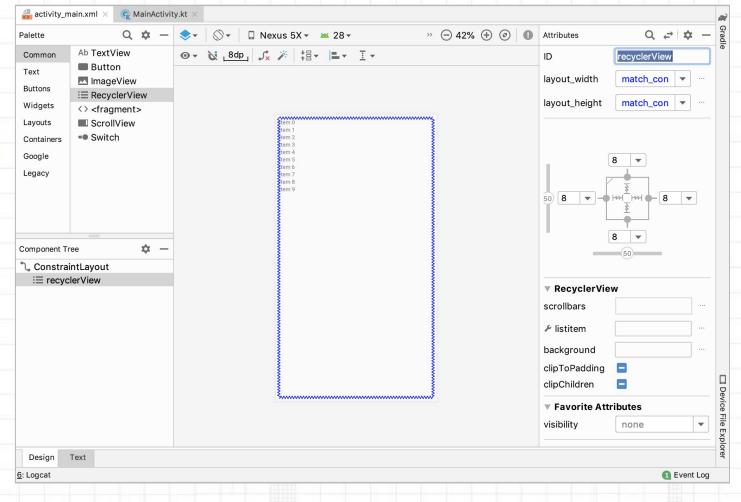
Kotlin



#### ทำการ New Project ขึ้นมาใหม่เป็น Kotlin ขึ้นมา ตั้งชื่อ Project ว่าอะไรก็ได้







ทำการดาวน์โหลด RecyclerView มาใช้งานหลังจากนั้นให้นำ Widget ไปวาง ทำการเชื่อม ConstraintLayout ทุกมุม ตั้งชื่อ ID ของ RecyclerView ว่า recyclerView



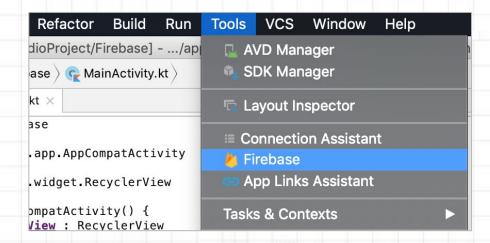
ทำการไปที่ MainActivity.kt ประกาศตัวแปรที่ Global Variable ใต้ Class ด้วยคำสั่ง:

lateinit var recyclerView : RecyclerView

ไปที่ onCreate() ทำการ ประกาศตัวแปรรับ findViewByld ที่ acitivy\_main.xml

recyclerView = findViewById(R.id.recyclerView)

ไปที่ เมนู Tools -> Firebase เพื่อเปิดใช้งาน Firebase Assistance เปิดเลือก Realtime Database กดที่ Save and retrieve data



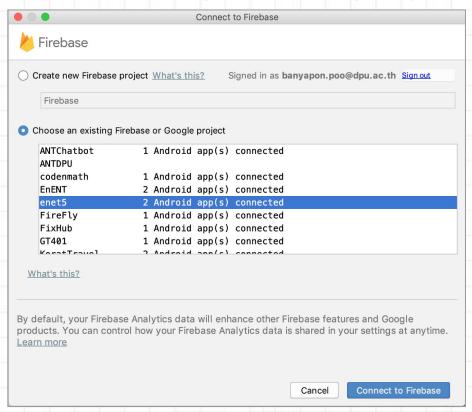
#### Realtime Database

Store and sync data in realtime across all connected clients. More info

Save and retrieve data



#### เลือก Choose an Existing Firebase or Google Project ของเราที่เคยสร้างไว้

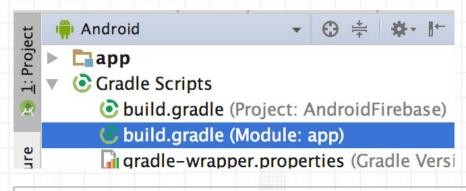


#### ทำตาม Step ของ Assistant ข้อ 1 และ ข้อ 2

- Connect your app to Firebase
  - Connected
- Add the Realtime Database to your app
  - Opendencies set up correctly



แก้ไข Module App ของ Gradle ใหม่เล็กน้อย:



```
dependencies {
    implementation fileTree(dir: 'libs', include: ['*.jar'])
    implementation "org.jetbrains.kotlin:kotlin-stdlib-jdk7:$kotlin_version"
    implementation 'com.android.support:appcompat-v7:28.0.0'
    implementation 'com.android.support.constraint:constraint-layout:1.1.3'
    implementation 'com.google.firebase:firebase-database:16.0.1'
    implementation 'com.google.firebase:firebase-core:16.0.1'
    testImplementation 'junit:junit:4.12'
    androidTestImplementation 'com.android.support.test:runner:1.0.2'
    androidTestImplementation
'com.android.support.test.espresso:espresso-core:3.0.2'
    implementation 'com.android.support:recyclerview-v7:28.0.0'
    implementation 'com.android.support:cardview-v7:28.0.0'
    implementation 'com.squareup.picasso:picasso:2.5.2'
}
```

เราจะใช้ Firebase Database ในการดึงข้อมูลมาแสดงผ่าน Adapter โดย ใช้ RecyclerView และ CardView มารับค่าข้อมูล และใช้ Picasso จัดการการเก็บแคชของข้อมูล



ไปที่ Gradle ส่วนของ Project ทำการแก้ไข Class Path dependencie ดังนี้:



build.gradle (Project: iApplication)

w build.gradle (Module: app)

4 gradle wrapper properties (Gradle Verd

```
buildscript {
    ext.kotlin_version = '1.3.11'
    repositories {
        google()
        jcenter()
}
dependencies {
        classpath 'com.android.tools.build:gradle:3.3.1'
        classpath "org.jetbrains.kotlin:kotlin-gradle-plugin:$kotlin_version"
        classpath 'com.google.gms:google-services:4.0.0'

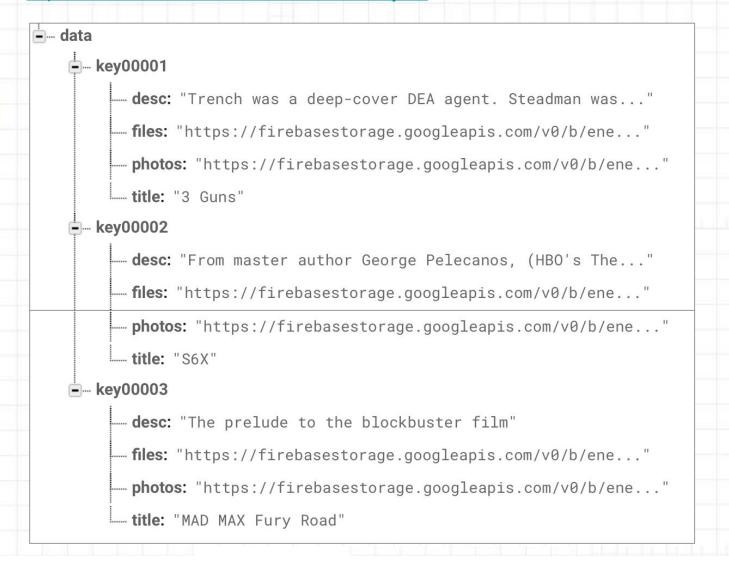
// NOTE: Do not place your application dependencies here; they belong
// in the individual module build.gradle files
```

แก้ google-services ให้อยู่ใน เวอร์ชันที่เสถียรคือ 4.0.0



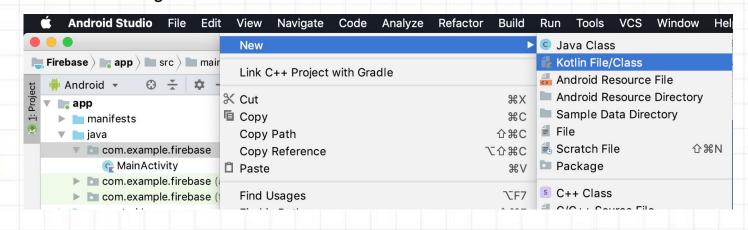
#### สำหรับโครงสร้างอยากให้ปรับโครงสร้างของ Firebase ให้เป็นตาม URL: ดังนี้:

#### https://enet5-7f9f6.firebaseio.com/bookshelf.json

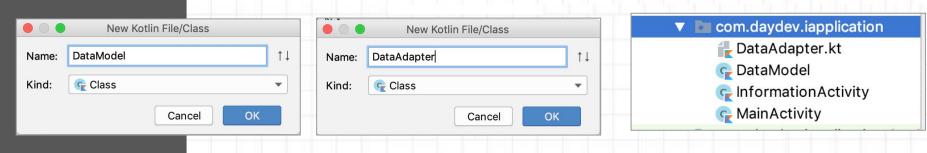




#### คลิกขวาที่ Package ของเราเลือก New -> Kotlin File/Class



#### ตั้งประเภทเป็น Class ตั้งชื่อว่า DataModel, DataAdapter ตรวจสอบตำแหน่ง Class

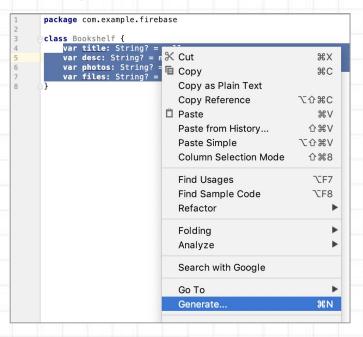


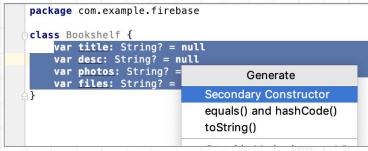
#### ประกาศตัวแปรใน Classของ DataModel ดังนี้ (สังเกตุมันคือ Key ของ Firebase ของเรา)

```
class DataModel {
  var title: String?=null
  var desc: String?=null
  var photos: String?=null
  var files: String?=null
  var key:String?=null
}
```



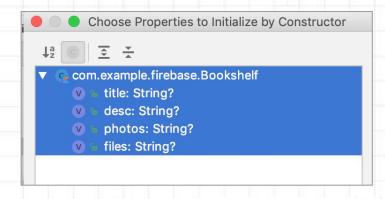
ลาก Drag Over ครอบทั้ง 4 ตัวแปร หลังจากนั้นคลิกขวาเลือก Generate...





เลือก Secondary Constructor เพื่อสร้าง Constructor

Constructor คือ สิ่งที่มีไว้สำหรับกำหนดค่าเริ่มต้นให้กับตัวแปร Class ของ object Constructor จะทำงานอัตโนมัติเมื่อมีการ new object ขึ้นมา



ทำการเลือกทั้งหมดเพื่อ Initialize ตัว Constructor อย่าลิมทำ none argument constructor

constructor()



#### Class ของ Bookshelf จะเป็นดังนี้

```
class DataModel {
    var title: String?=null
    var desc: String?=null
    var photos: String?=null
    var files: String?=null
    var key:String?=null

constructor()

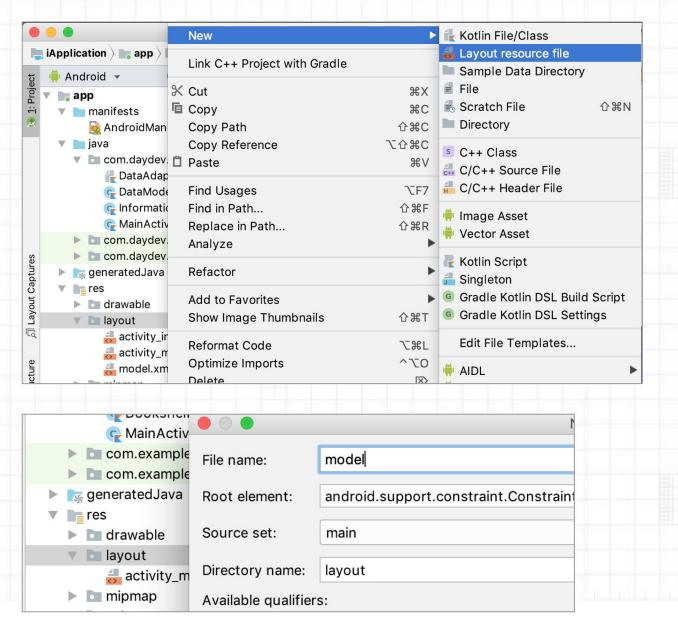
constructor(title: String?, desc: String?, photos: String?, key: String?, files: String?) {
    this.title = title
    this.desc = desc
    this.photos = photos
    this.key = key
    this.files = files
}
```

#### เพิ่มฟังก์ชันใหม่ชื่อว่า toMap() เข้าไปก่อน ปิด Class เพื่อเป็นการ Map ค่า Key จาก Firebase เข้าไปยัง constructor ของเรา

```
fun toMap(): Map<String, Any> {
  val result = HashMap<String, Any>()
  result.put("title", title!!)
  result.put("desc", desc!!)
  result.put("thumbnail", photos!!)
  result.put("key", key!!)
  result.put("files", files!!)
  return result
}
```



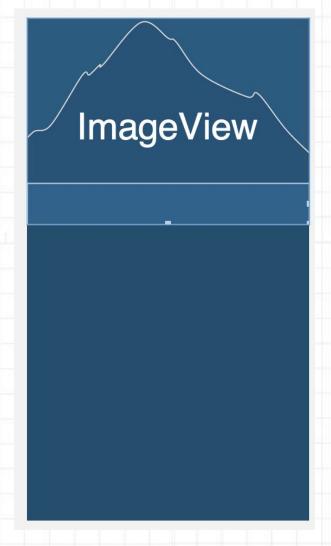
#### สร้าง Layout Resource File ใหม่ใน res->layout ชื่อว่า model





#### ไฟลุ Layout จะมี XML ดังนี้:

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
xmlns:android="http://schemas.android.com/apk/res/a
ndroid"
       android:layout_width="match_parent"
       android:layout_height="300dp"
       android:orientation="horizontal"
 <LinearLayout
      android:layout_width="match_parent"
      android:layout_height="match_parent"
      android:orientation="vertical">
   <ImageView
        android:id="@+id/thumbnail"
        android:layout_width="match_parent"
        android:layout_height="240dp"
        android:scaleType="centerCrop"/>
   <TextView
        android:id="@+id/title"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_weight="1"
        android:textAlignment="center"
android:textColor="@color/cardview_dark_background
        android:textSize="16sp"/>
 </LinearLayout>
</LinearLayout>
```





ไปที่ DataAdapter เราจะประกาศฟังก์ชัน ในรอบนี้จะทำงานเรียก DataModel มา ปรากฏ และใช้งาน ViewHolder มาจัดการ เช่นเดิม และรอบนี้เราจะเรียก List ของ Array ทั้งก้อนมาใส่ โดยใช้ Pattern จาก Class DataModel

```
class DataAdapter(val dataModelList: List<DataModel>) : RecyclerView.Adapter<ViewHolder>() {
 override fun onCreateViewHolder(p0: ViewGroup, p1: Int): ViewHolder {
 override fun onBindViewHolder(p0: ViewHolder, p1: Int) {
 override fun getItemCount(): Int {
class ViewHolder(itemView: View) : RecyclerView.ViewHolder(itemView) {
 var textTitle: TextView = itemView.findViewByld(R.id.title)
 var imageView: ImageView
 init {
    imageView = itemView.findViewById(R.id.thumbnail)
```

ส่วนของ ViewHolder นั้นมีการประกาศ widget ของ ImageView และ TextView แล้ว ผ่านตัวแปร thumbnail, textTitle



#### ในเมธอด getItemCount() ให้ใส่ return items.size เพื่อรับจำนวน Arrays

```
override fun getItemCount(): Int {
    return dataModelList.size
}
```

ไปที่ เมธอด onCreateViewHolder() ให้ลบข้อมูล แล้วใส่ Code ต่อไปนี้:

```
override fun onCreateViewHolder(p0: ViewGroup, p1: Int): ViewHolder {
    return ViewHolder(LayoutInflater.from(p0.context).inflate(R.layout.model,p0,false))
}
```

ในเมธอด onBindViewHolder() ให้เราสร้างการ เรียกใช้ DataModel ไปยัดข้อมูลง ViewHolder ทีละแถว ส่วนของรูปภาพเราจัดการ Cache ด้วย Picasso



Picasso เป็น Third Party Library สำหรับการจัดการแคชบนรูปภาพ ImageView Cache มีการดาวน์โหลดรูปภาพภายหลัง Activity เริ่มทำงานโดยเก็บ cache ให้ทันที สามารถใช้งานได้ง่ายทั้งแบบ jar ไว้ใน Project หรือ Build ผ่าน Gradle ตรงๆ (ข้อมูล เพิ่มเติม: <a href="http://square.github.io/picasso/">http://square.github.io/picasso/</a>)







Network

Disk

Memory



#### ไปที่ MainActivity.kt ประกาศตัวแปรเพิ่ม

```
private val TAG = "Comic"
private lateinit var response_data: MutableList<DataModel>
private var dataAdapter: DataAdapter? = null
private lateinit var recyclerView: RecyclerView
```

**private lateinit var firebaseDatabase**: FirebaseDatabase **private lateinit var databaseReference**: DatabaseReference

### ทำการสร้าง LayoutGridManager เป็น 2 ประกาศตัวแปร response\_data เป็น mutableList() เป็น ListArray รูปแบบหนึ่ง

```
recyclerView = findViewByld(R.id.recyclerView)
recyclerView!!.layoutManager = LinearLayoutManager(this)
recyclerView!!.setLayoutManager(GridLayoutManager(this, 2))

firebaseDatabase = FirebaseDatabase.getInstance()
databaseReference = firebaseDatabase!!.getReference("bookshelf/data")
response_data = mutableListOf()

dataAdapter = DataAdapter(response_data as ArrayList<DataModel>)
recyclerView!!.setAdapter(dataAdapter)
bindingData()
```

#### ทำการ Create method ใหม่

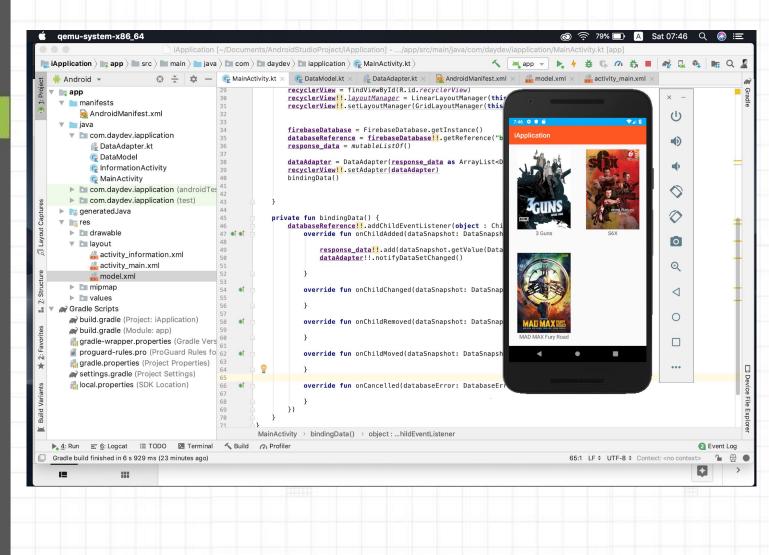


จะเห็นว่ามีการเรียกไปยัง refernce Database ส่วนของ Node ที่ชื่อว่า bookshelf/data ให้ ตรงกับส่วนที่เราเก็บข้อมูลไว้ใน Web CMS เมื่อเสร็จขั้นตอนการ setAdapter ของ recyclersView แล้วเราจะต้องสร้าง Method ใหม่ชื่อว่า bindingData();

```
private fun bindingData() {
 databaseReference!!.addChildEventListener(object : ChildEventListener {
    override fun onChildAdded(dataSnapshot: DataSnapshot, s: String?) {
      response_data!!.add(dataSnapshot.getValue(DataModel::class.java)!!)
      dataAdapter!!.notifyDataSetChanged()
   override fun onChildChanged(dataSnapshot: DataSnapshot, s: String?) {
    override fun onChildRemoved(dataSnapshot: DataSnapshot) {
    override fun onChildMoved(dataSnapshot: DataSnapshot, s: String?) {
    override fun onCancelled(databaseError: DatabaseError) {
```



#### ทดสอบแอพพลิเคชันของเรา





# THANK YOU FOR YOUR ATTENTIONS