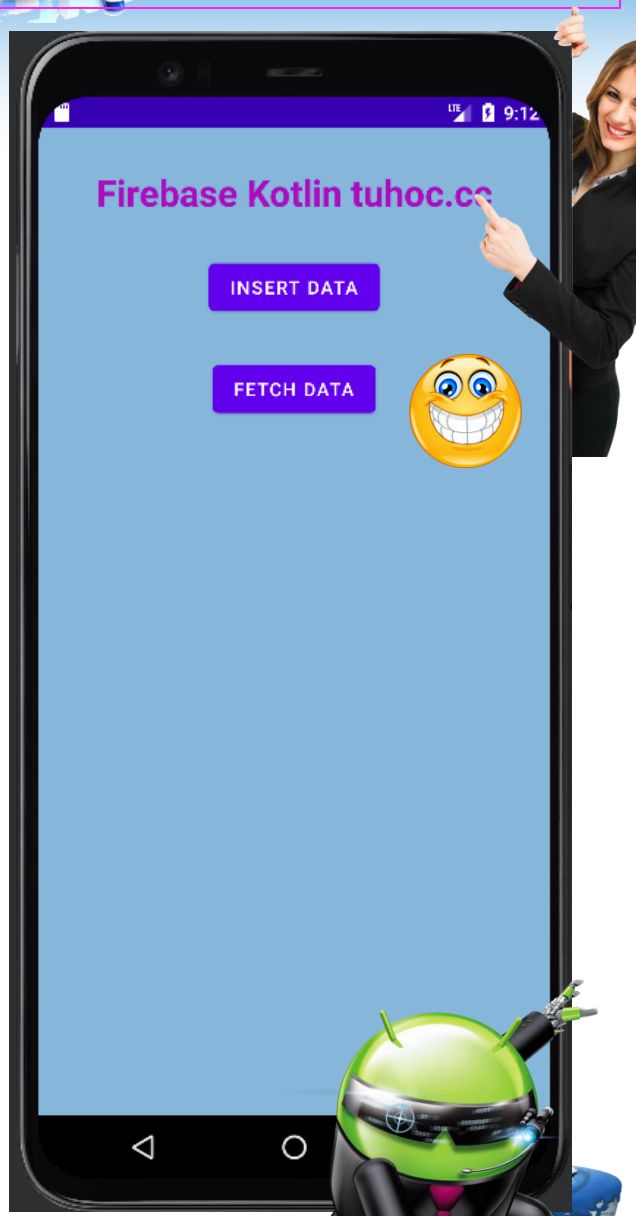
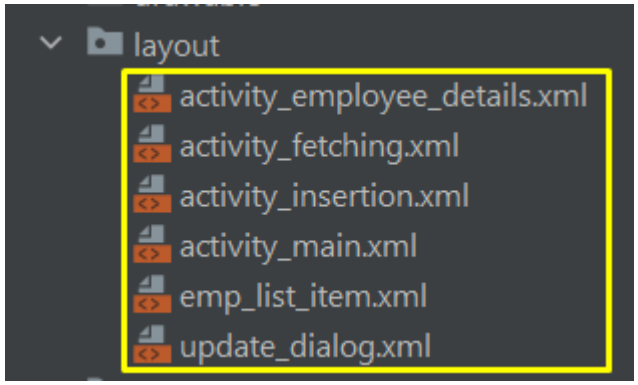
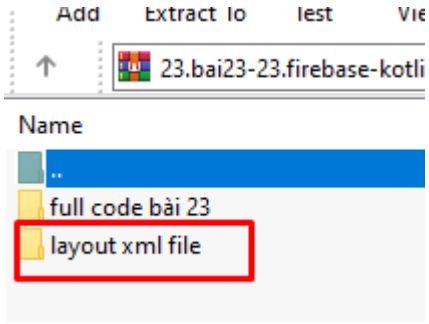


CRUD firebase kotlin

Firebase Database with Kotlin

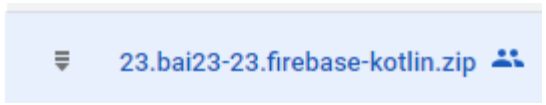
❑ 0 . File xml layout và code

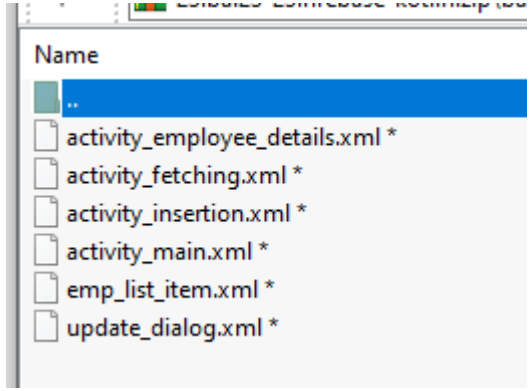




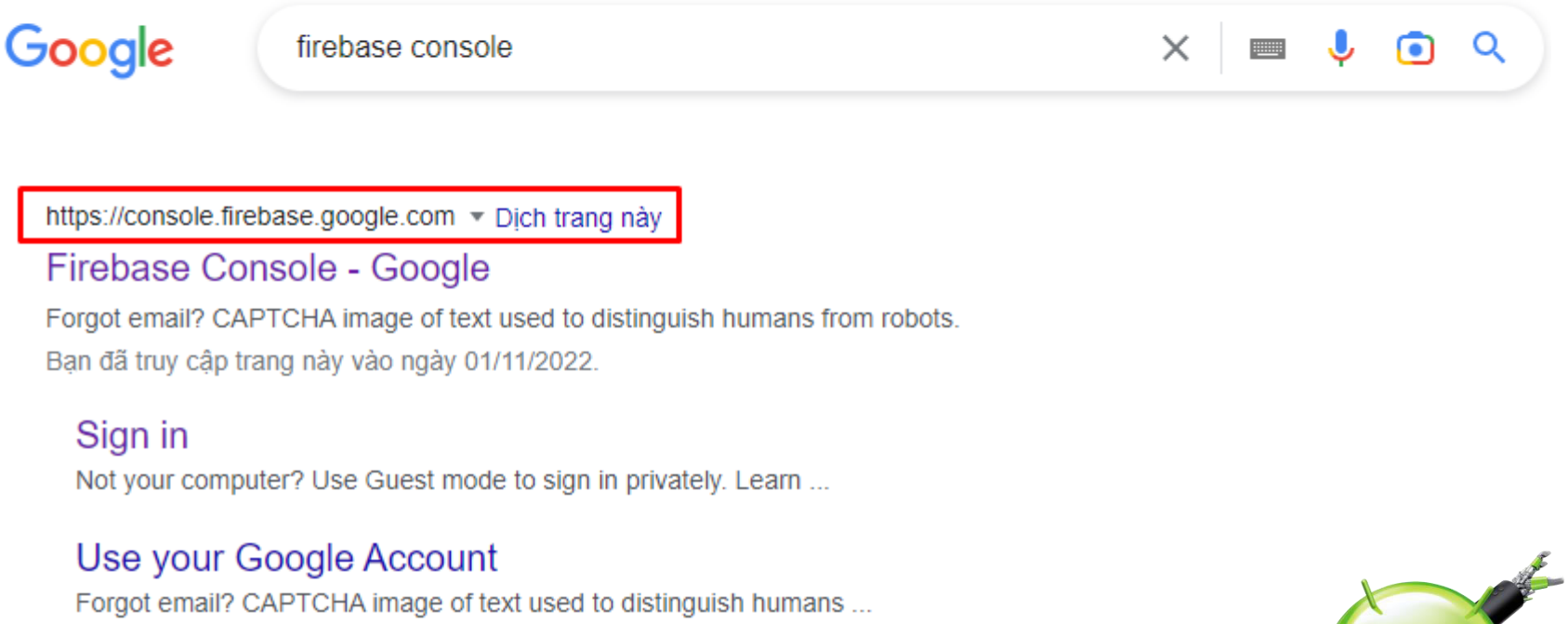
File xml layout
và code toàn bộ bài 23
Mật khẩu giải nén: firebase

http://android.tuhoc.cc/



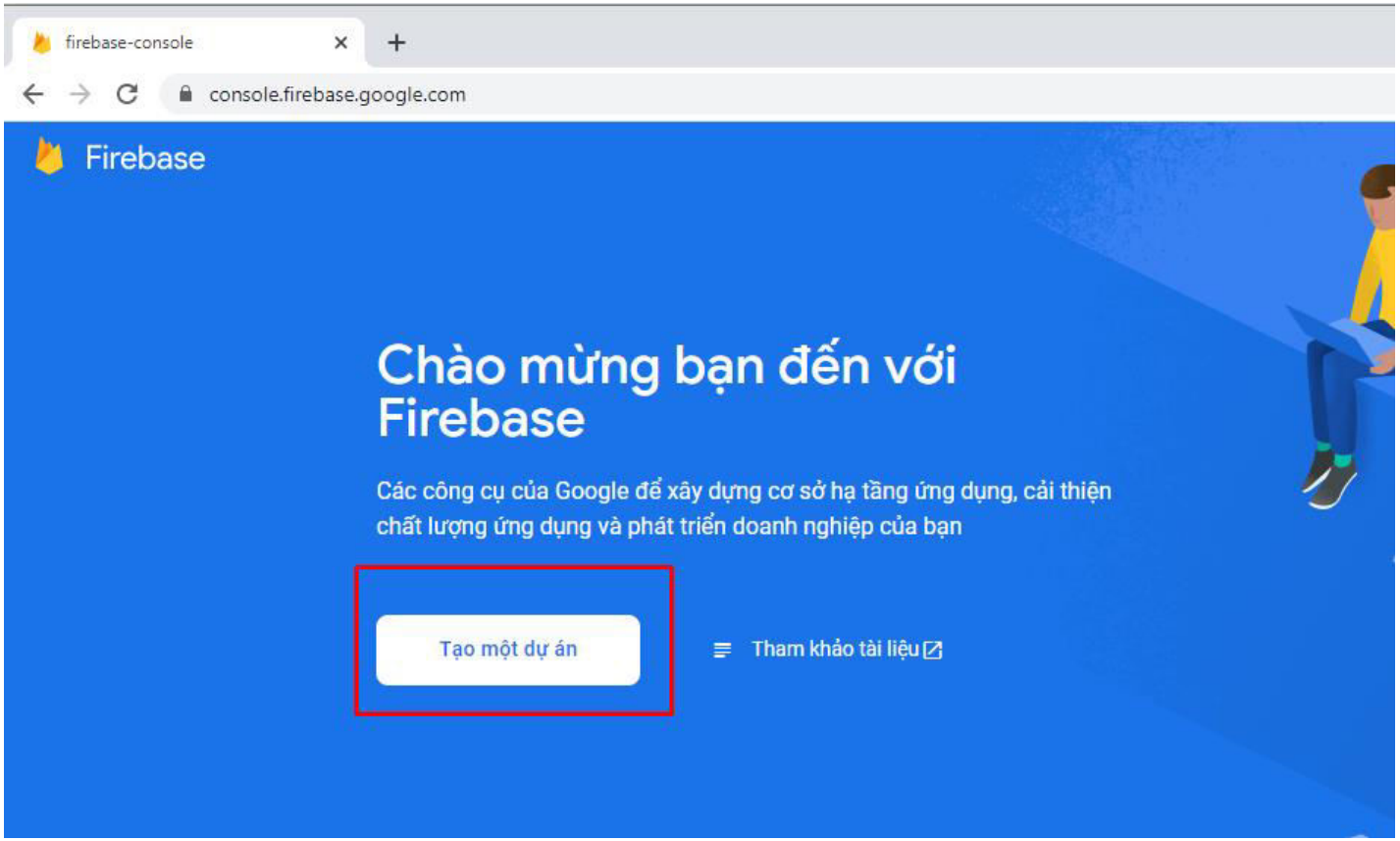


❑ 1. Truy cập firebase :



❑ 2. Tạo dự án :

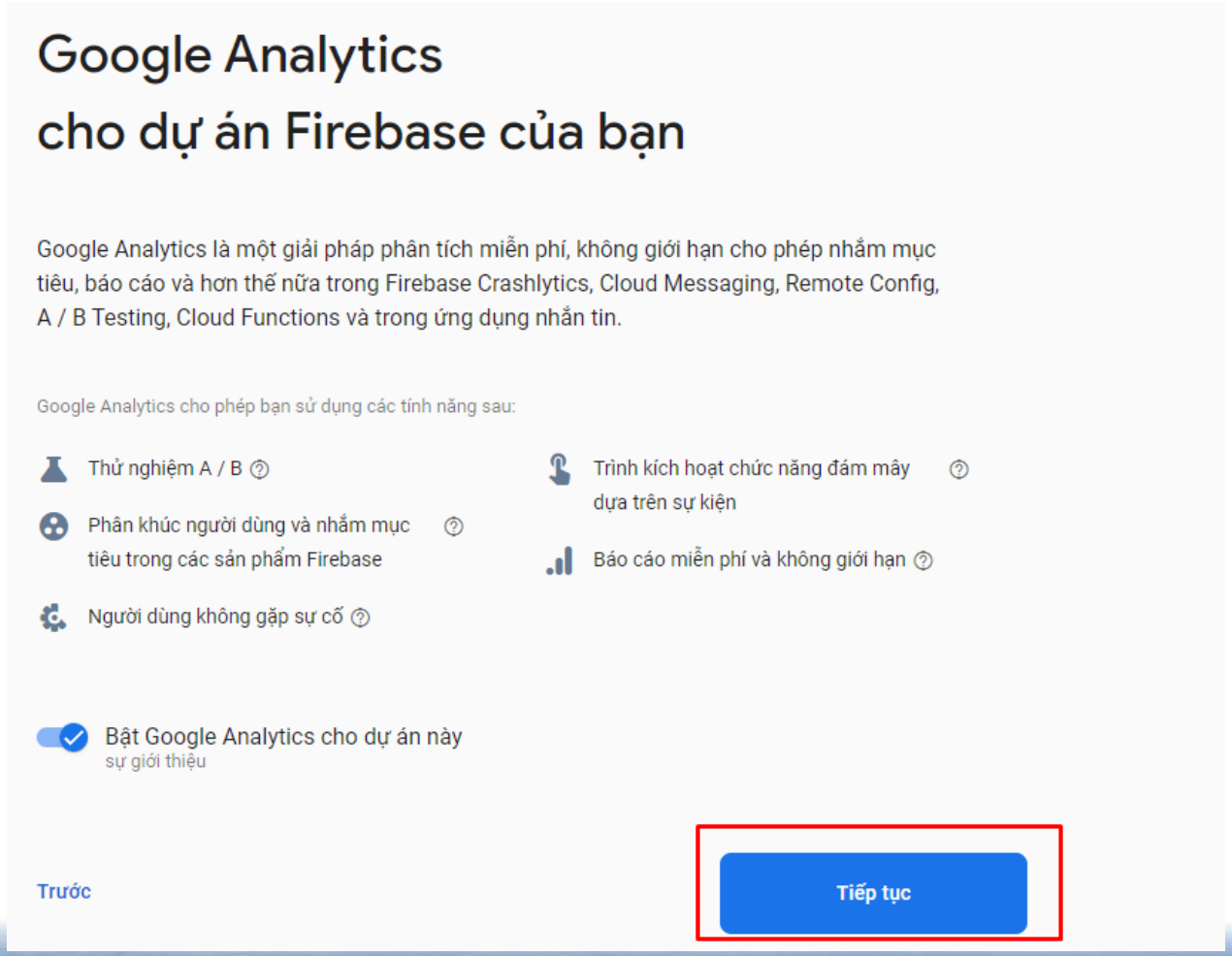
✓ *Cần đăng nhập gmail trước khi tạo :*



❑ 2. Tạo dự án :

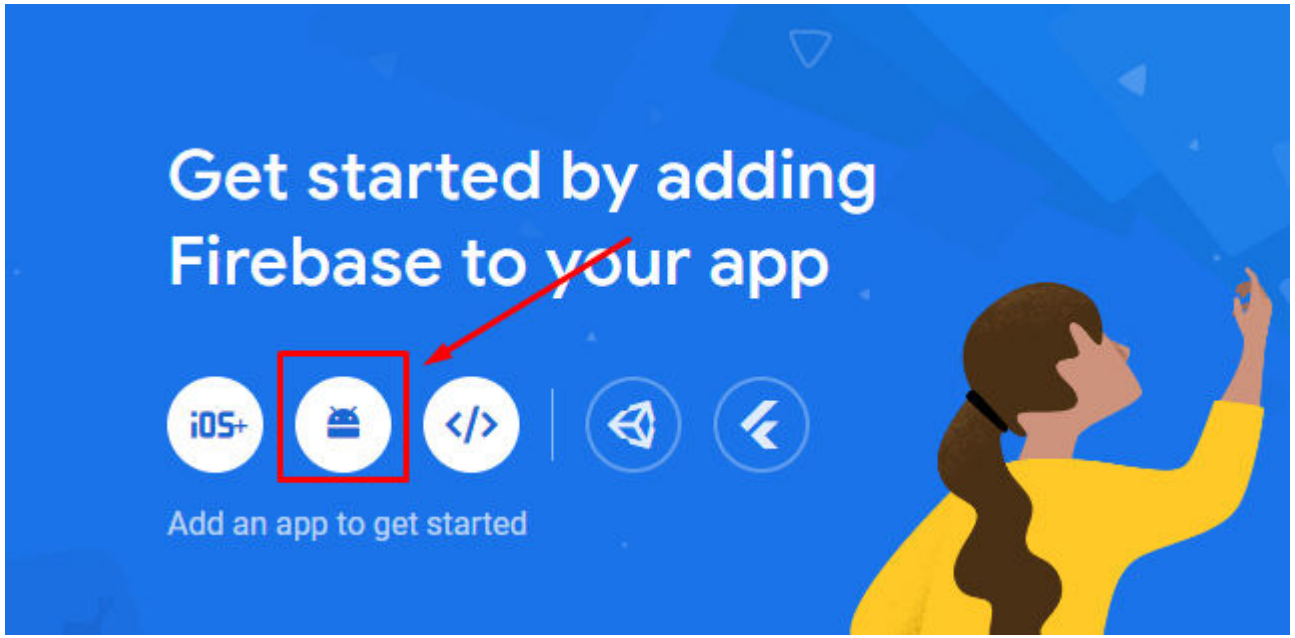

Hãy bắt đầu bằng cách đặt tên cho dự án của bạn
tên của dự án
tuhoc-Firebase
my-awesome-project-id
Tôi đồng ý với Điều khoản dịch vụ của Firebase
Tôi xác nhận rằng tôi sẽ chỉ sử dụng Firebase cho các hoạt động thương mại, công ty, sáng tạo hoặc nghề nghiệp của mình.
Tiếp tục

❑ 2. Tạo dự án:


Google Analytics
cho dự án Firebase của bạn
Google Analytics là một giải pháp phân tích miễn phí, không giới hạn cho phép nhắm mục tiêu, báo cáo và hơn thế nữa trong Firebase Crashlytics, Cloud Messaging, Remote Config, A / B Testing, Cloud Functions và trong ứng dụng nhắn tin.
Google Analytics cho phép bạn sử dụng các tính năng sau:
Thử nghiệm A / B
Phân khúc người dùng và nhắm mục tiêu trong các sản phẩm Firebase
Người dùng không gặp sự cố
Trình kích hoạt chức năng đám mây dựa trên sự kiện
Báo cáo miễn phí và không giới hạn
Bật Google Analytics cho dự án này
sự giới thiệu
Trước
Tiếp tục

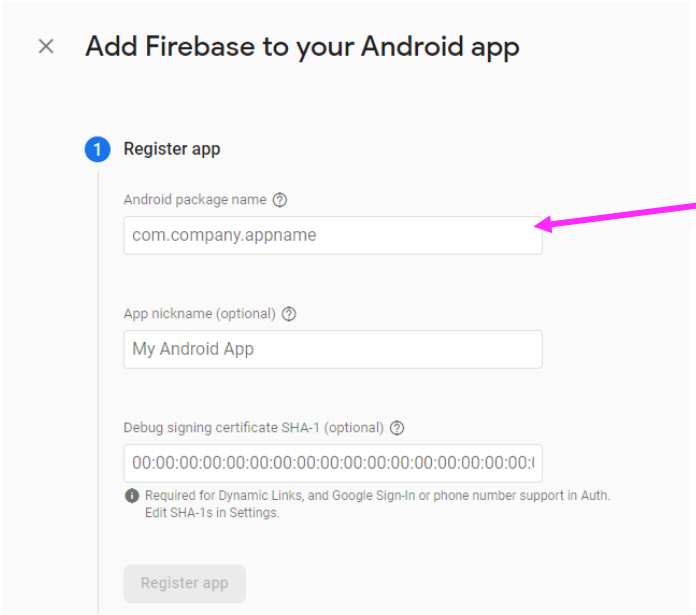
❑ 2. Tạo dự án :

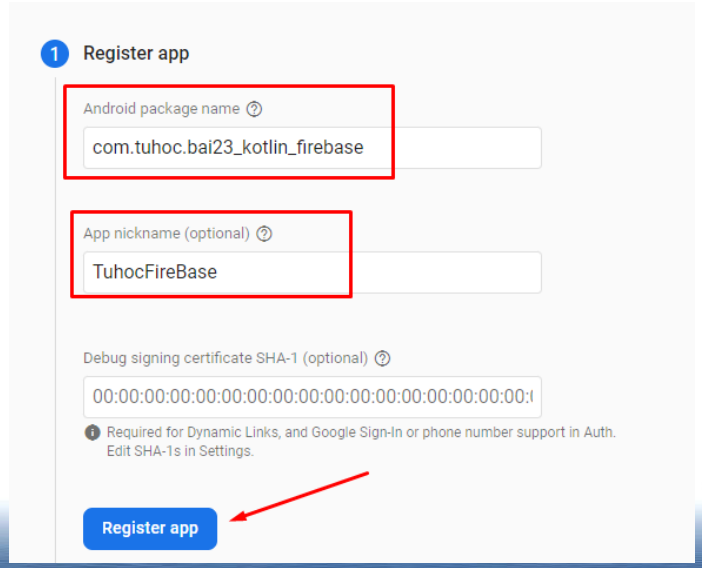
✓ *Chọn android :*



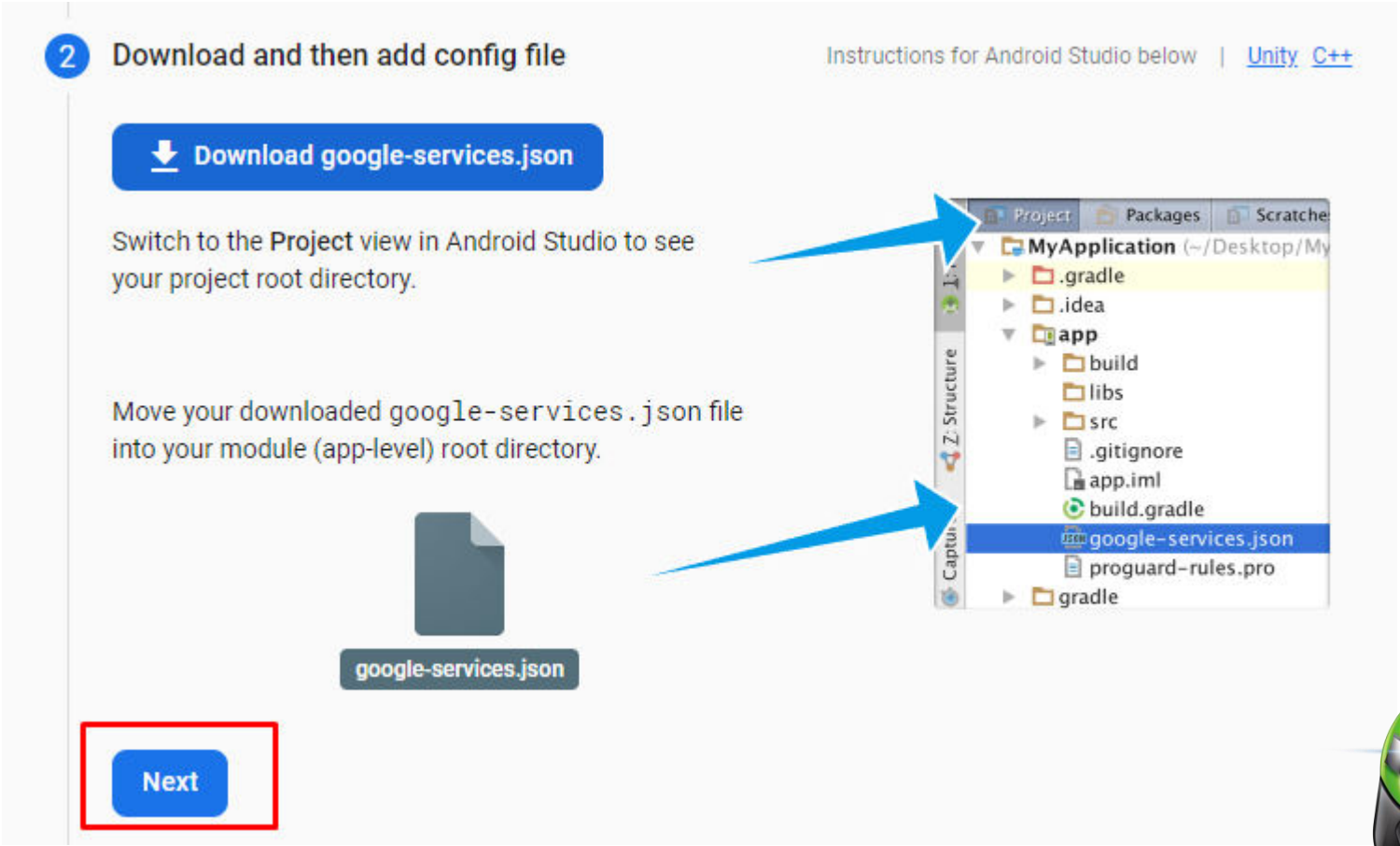
❑ 2. Tạo dự án :

✓ ***Register app :***

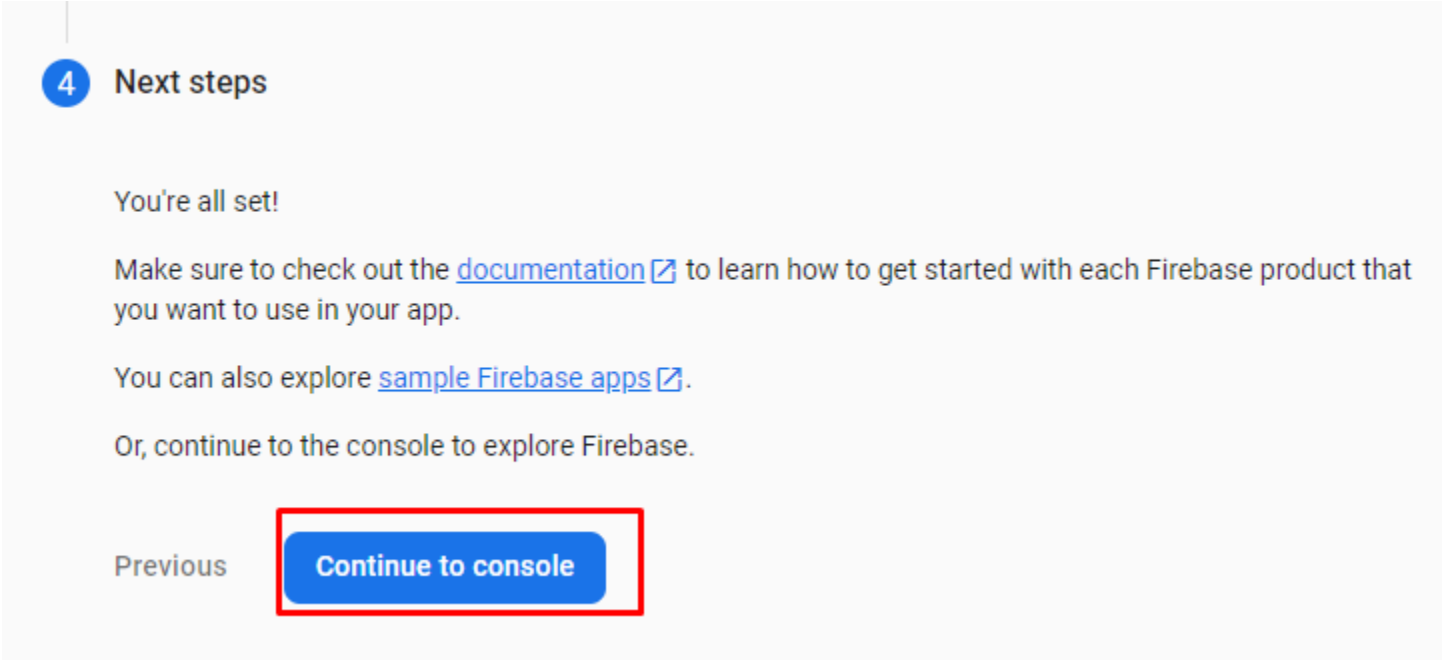




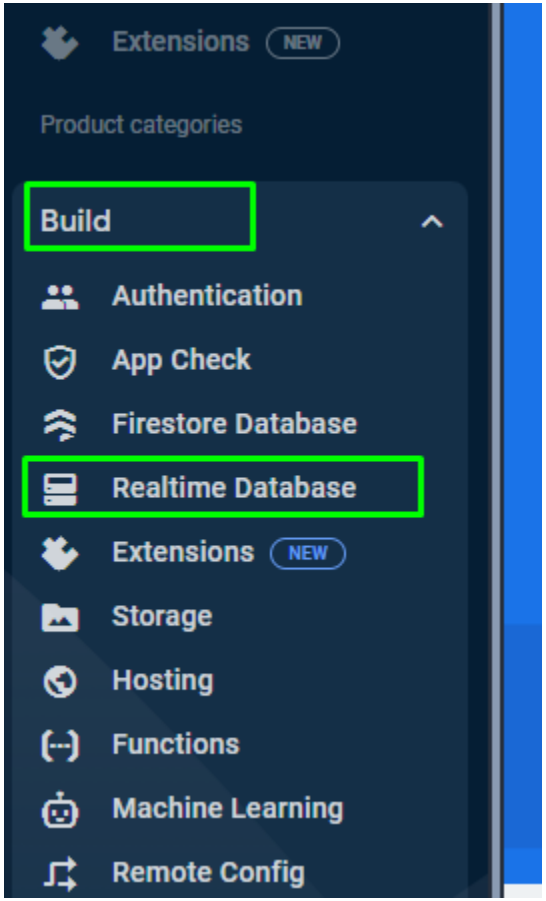
❑ 2. Tạo dự án :

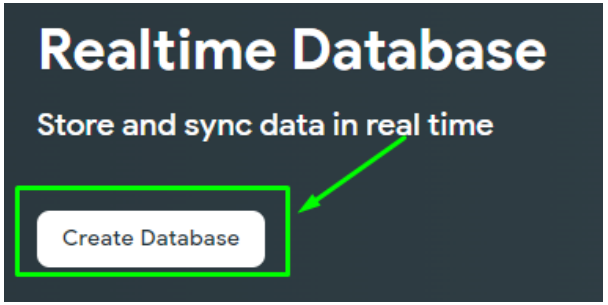

2
Download and then add config file
Instructions for Android Studio below | Unity C++
Download google-services.json
Switch to the Project view in Android Studio to see your project root directory.
Move your downloaded google-services.json file into your module (app-level) root directory.
google-services.json
Next
Project
Packages
Scratche
MyApplication (~/Desktop/My
.gradle
.idea
app
build
libs
src
.gitignore
app.iml
build.gradle
google-services.json
proguard-rules.pro
gradle
Z: Structure

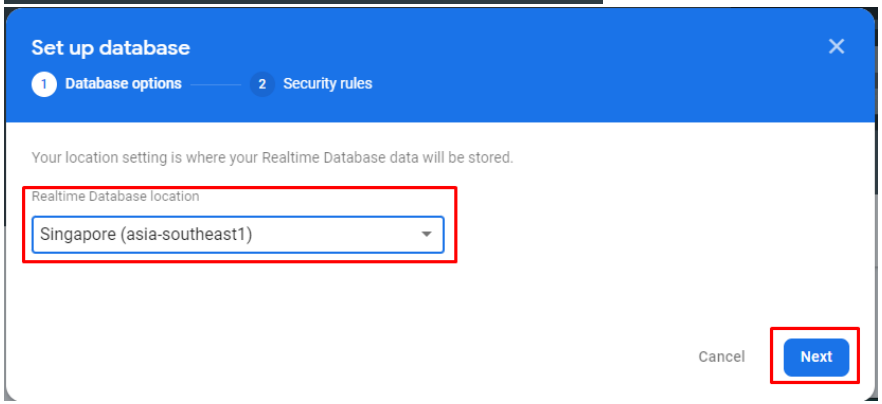
❑ 2. Tạo dự án :

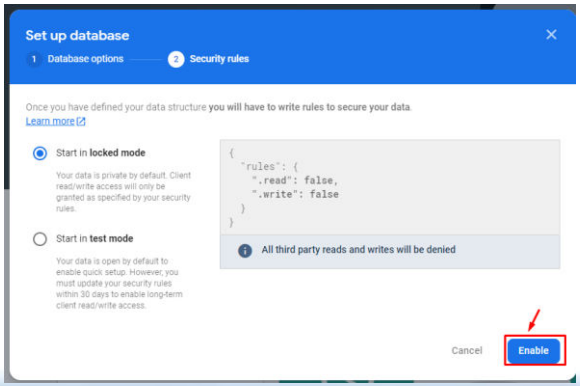

4
Next steps
You're all set!
Make sure to check out the documentation to learn how to get started with each Firebase product that you want to use in your app.
You can also explore sample Firebase apps.
Or, continue to the console to explore Firebase.
Previous
Continue to console

❑ 3. Firebase Realtime Database:



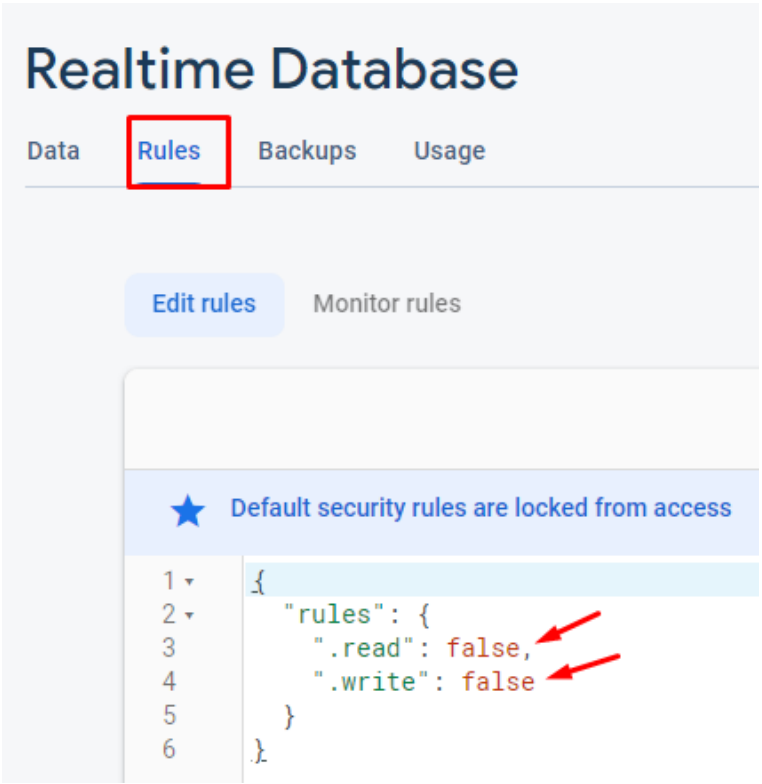


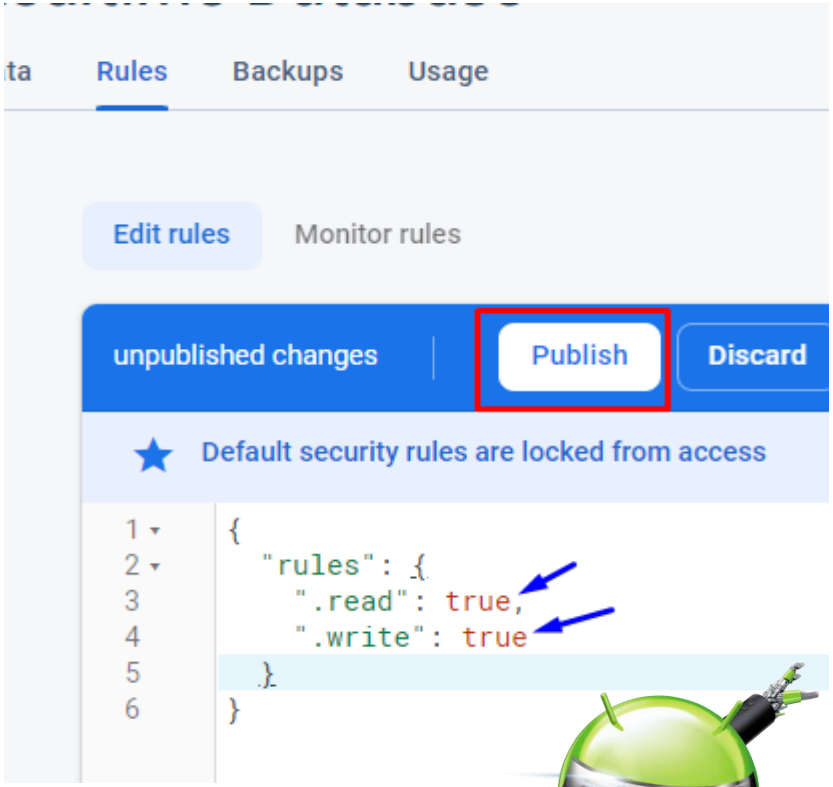




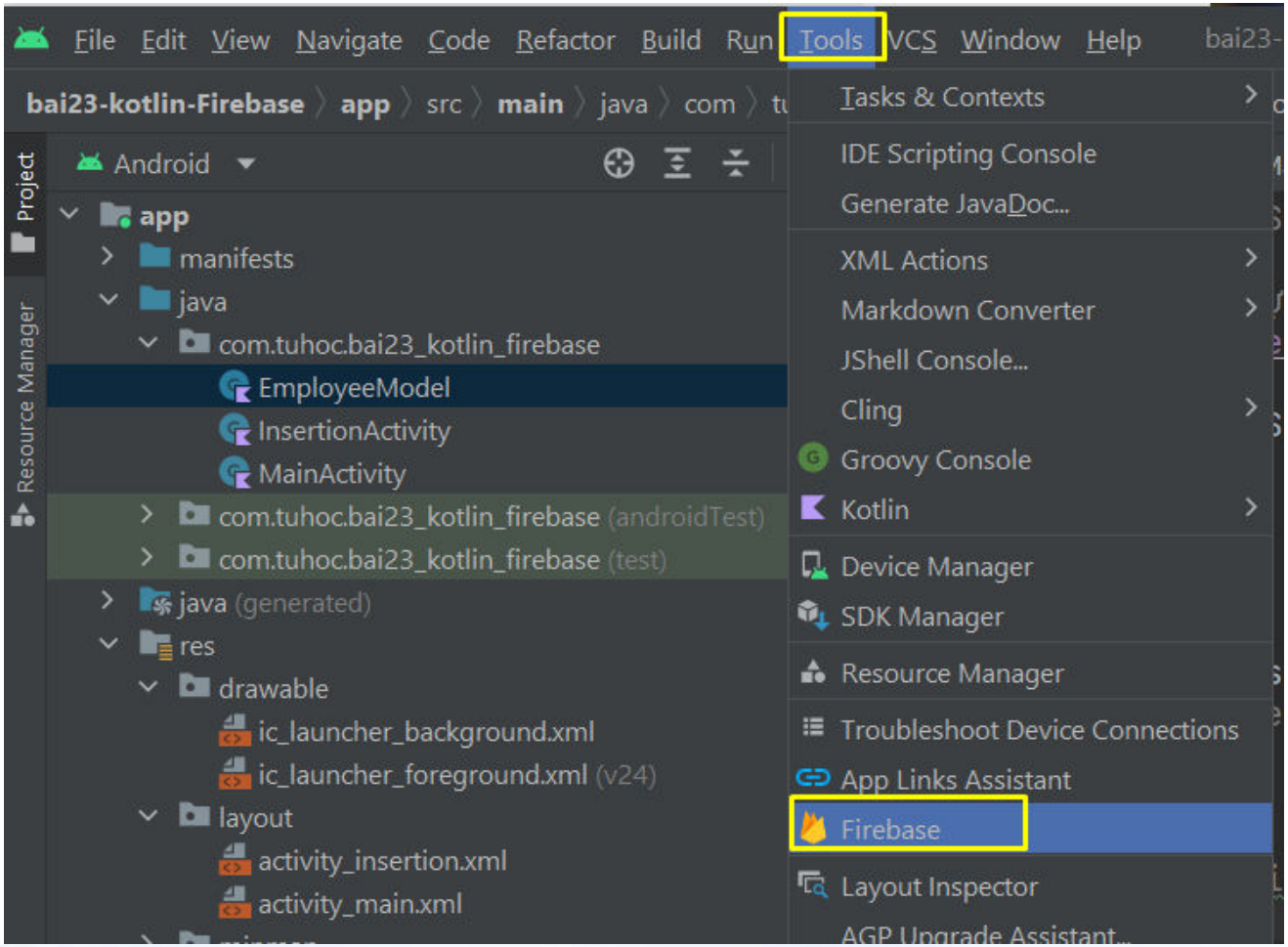
❑ 3. Firebase Realtime Database:

✓ *Set Rule for read and write database:*

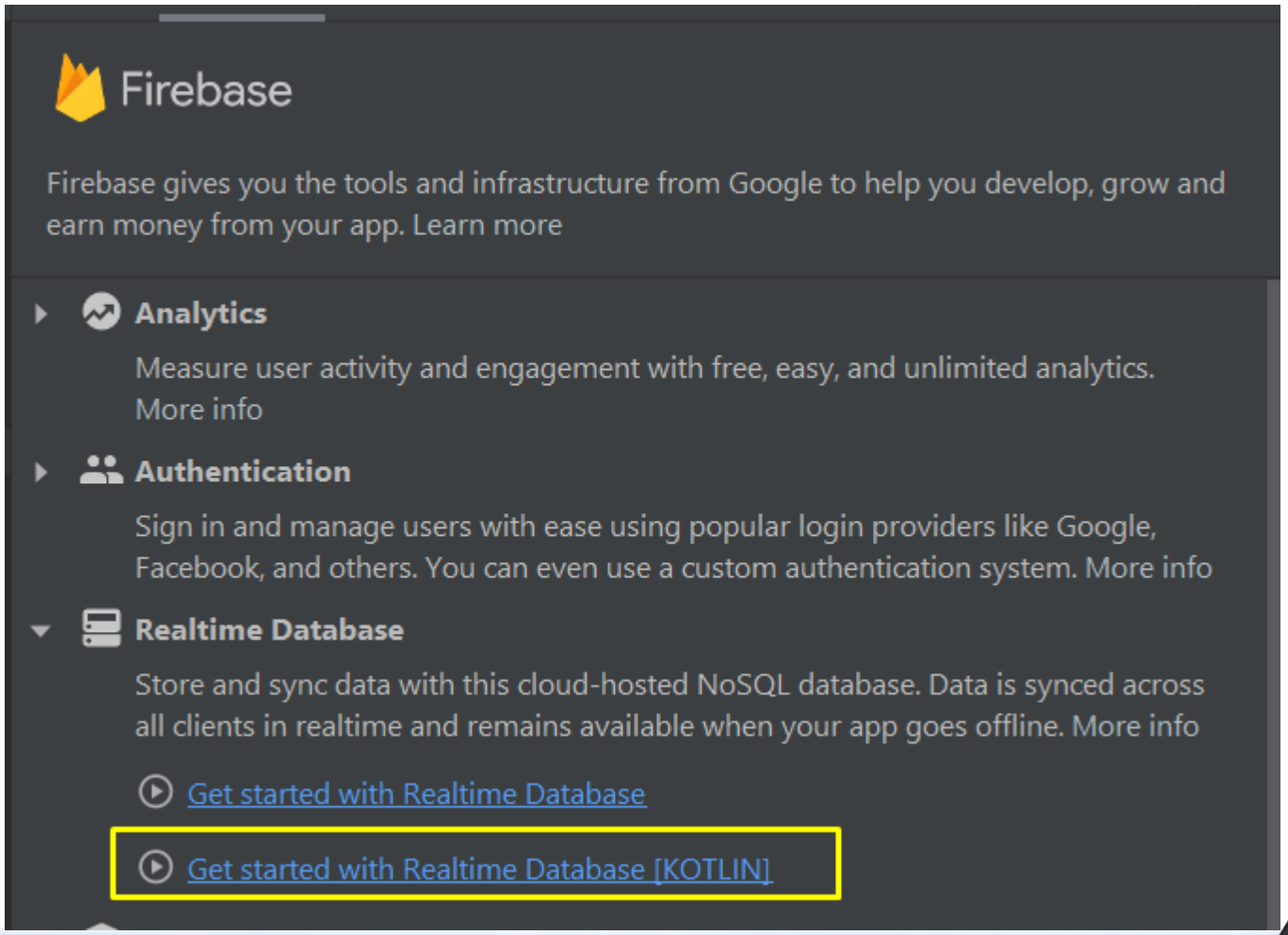




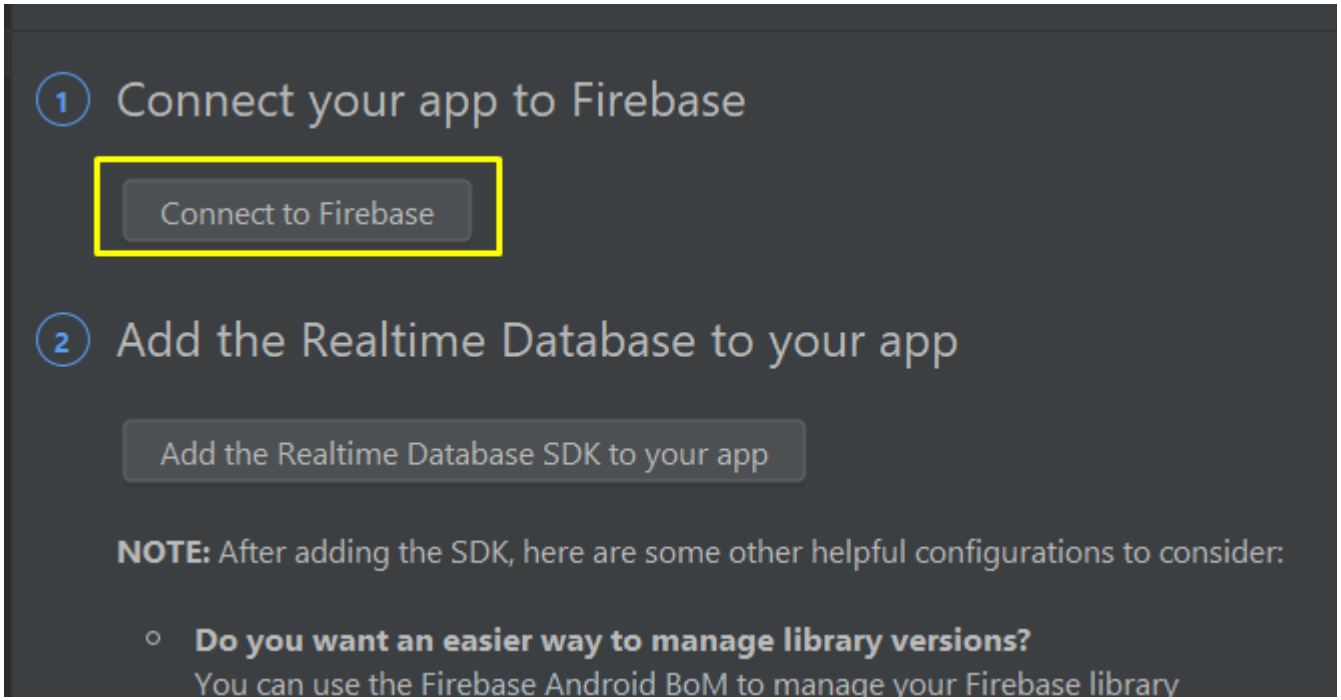
❑ 4. Connect firebase on AndroidStudio :

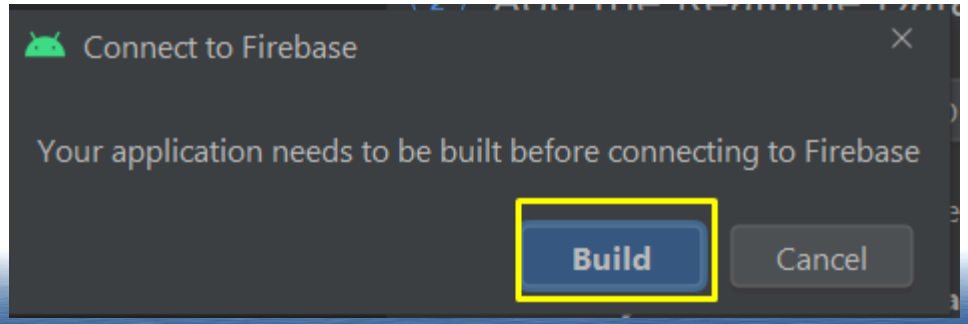


❑ 4. Connect firebase on AndroidStudio :

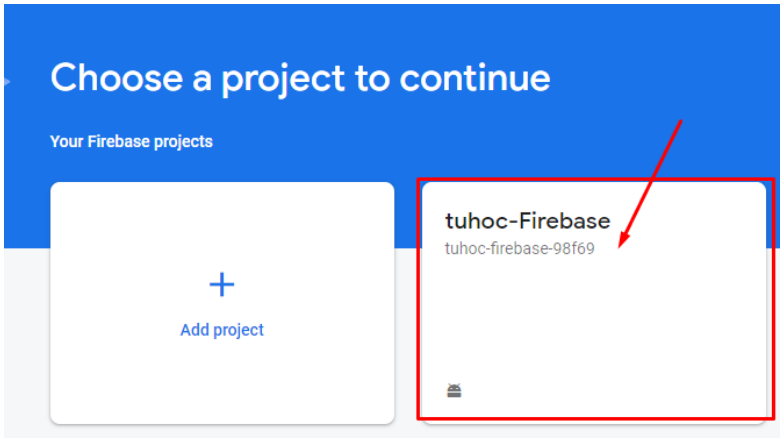


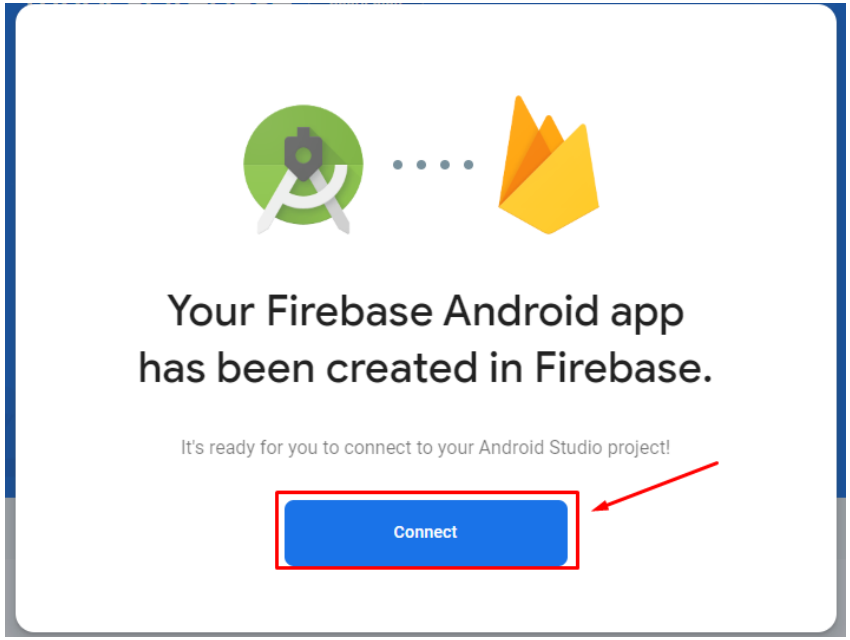
❑ 4. Connect firebase on AndroidStudio :

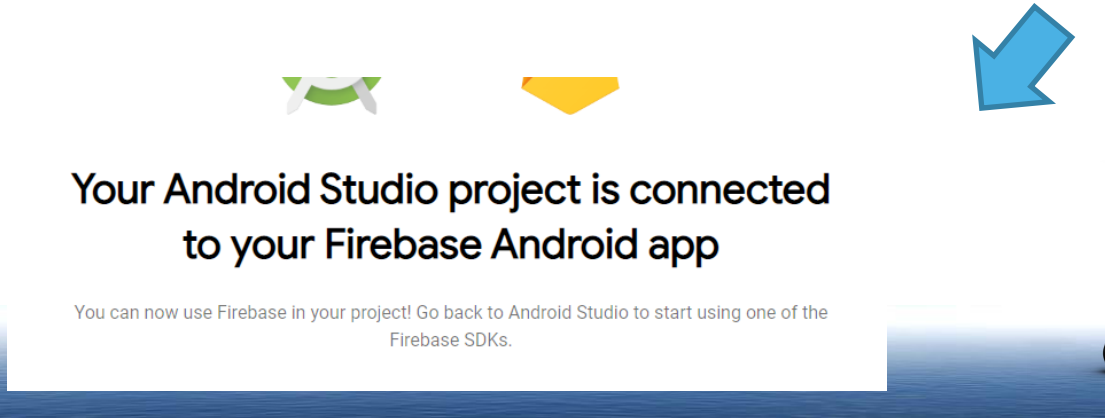




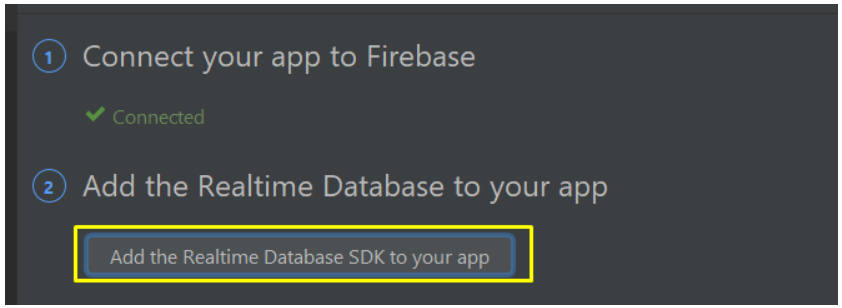
❑ 4. Connect firebase on AndroidStudio :

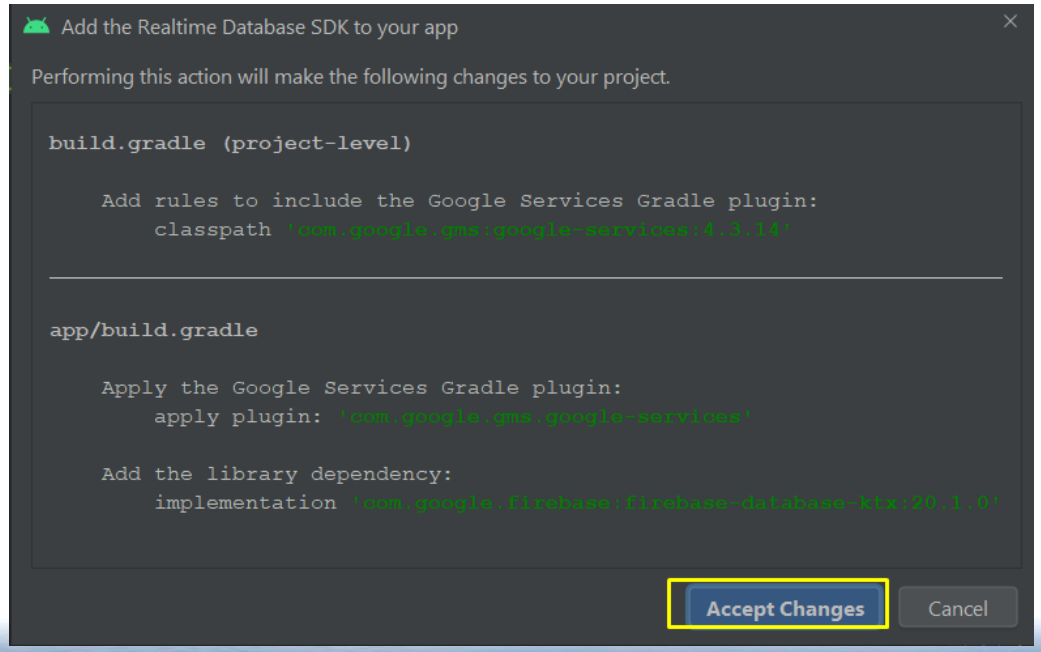


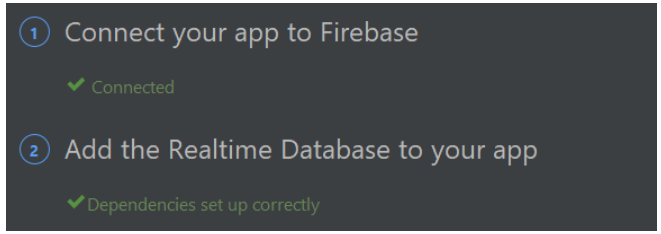




❑ 4. Connect firebase on AndroidStudio :

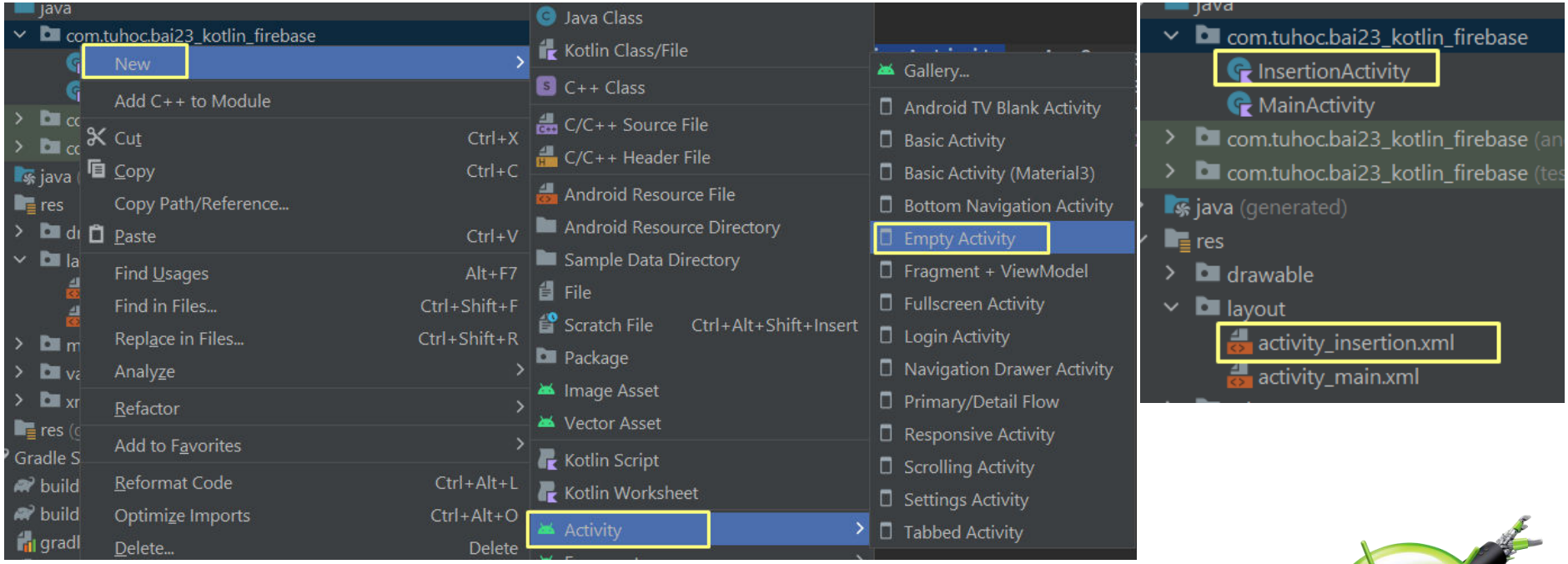






❑ 5. Tạo Activity insert data :

✓ *Xem lại bài 14, 14.2 về intent :*



❑ 5. Tạo Activity insert data :

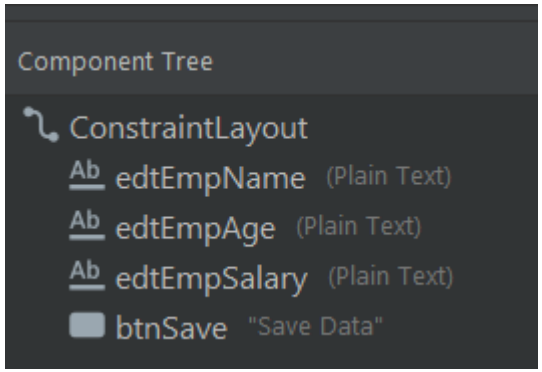
✓ *Code InsertionActivity :*

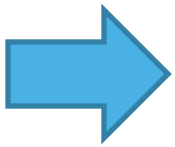
activity_main.xml | MainActivity.kt | activity_insertion.xml | InsertionActivity.kt

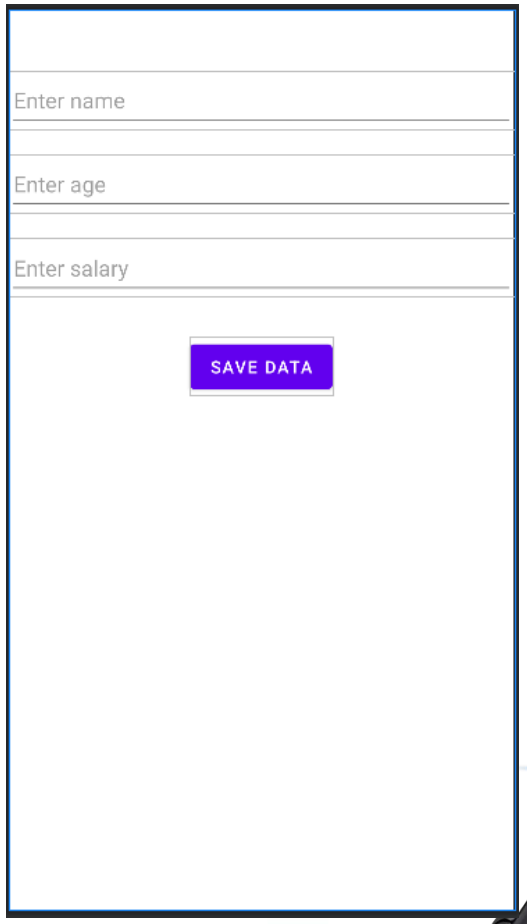
```
1   package com.tuhoc.bai23_kotlin_firebase
2
3   import ...
5
6   class InsertionActivity : AppCompatActivity() {
7       override fun onCreate(savedInstanceState: Bundle?) {
8           super.onCreate(savedInstanceState)
9           setContentView(R.layout.activity_insertion)
10      }
11  }
```

❑ 5. Tạo Activity insert data :

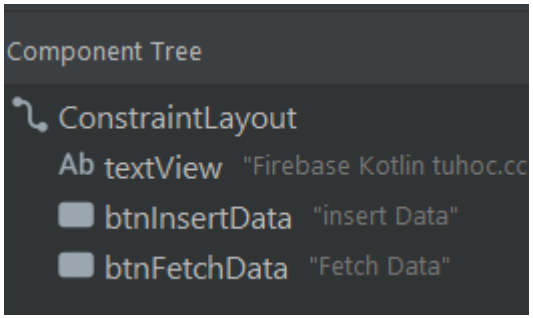
✓ *Giao diện Màn hình activity_insertion.xml :*

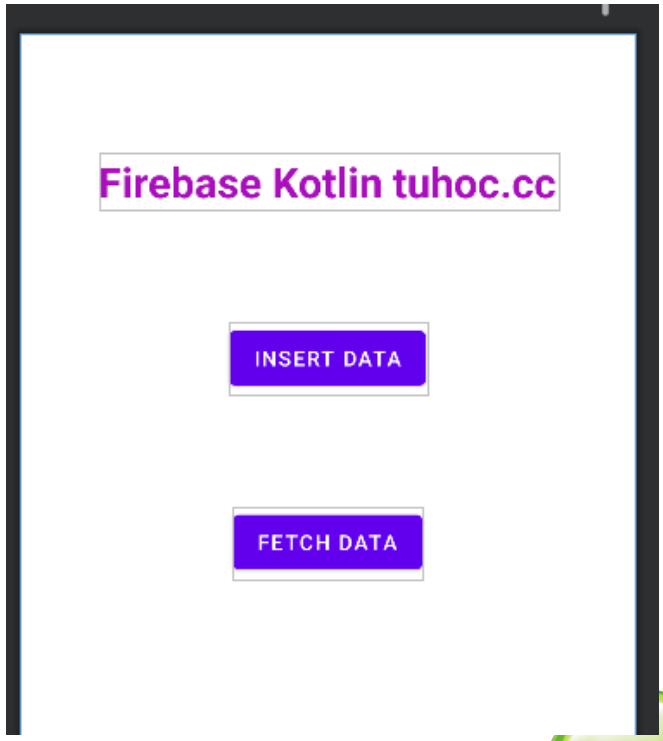






❑ 6. Code giao diện chính activity_main.xml :

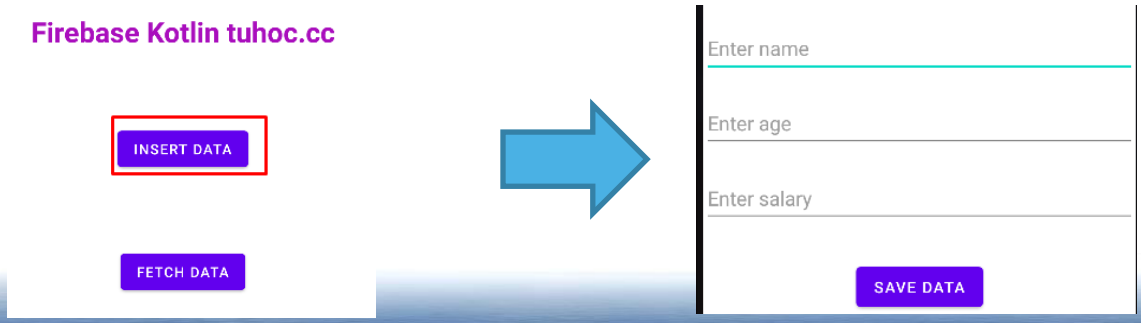




❑ 7. Code MainActivity :

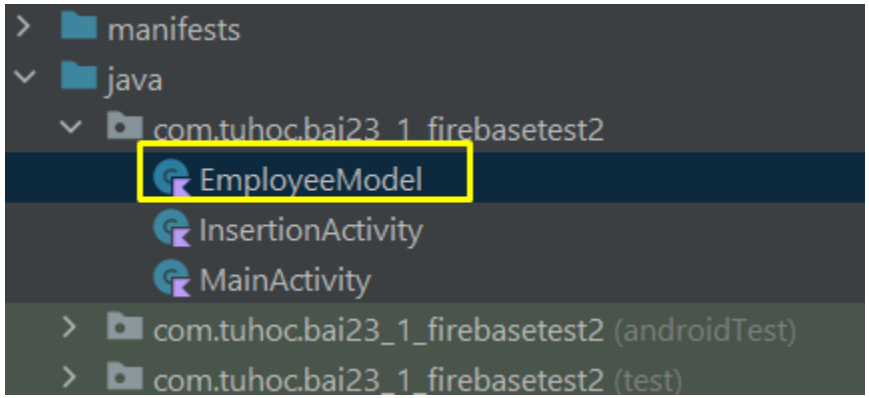
✓ *Gọi màn hình insertionActivity khi click vào button :*

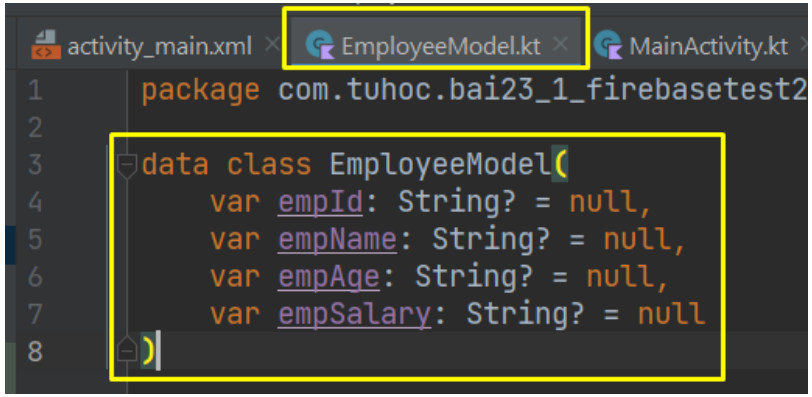
```
tivity_main.xml | MainActivity.kt | activity_insertion.xml | InsertionActivity.kt | build.gradle (:app) | build.gr

package com.tuhoc.bai23_kotlin_firebase

import ...

class MainActivity : AppCompatActivity() {
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_main)

        //code btninsertData mở màn hình insertionActivity
        btnInsertData.setOnClickListener {  it: View!
            val intent = Intent( packageContext: this,InsertionActivity::class.java)
            startActivity(intent)
        }
    }
}
```



❑ 8. add Class EmployeeModel:

✓ *Class sẽ gồm các thành phần trong table sau này cần thêm :*





❑ 8. Code InsertionActivity xử lý thêm dữ liệu vào database :

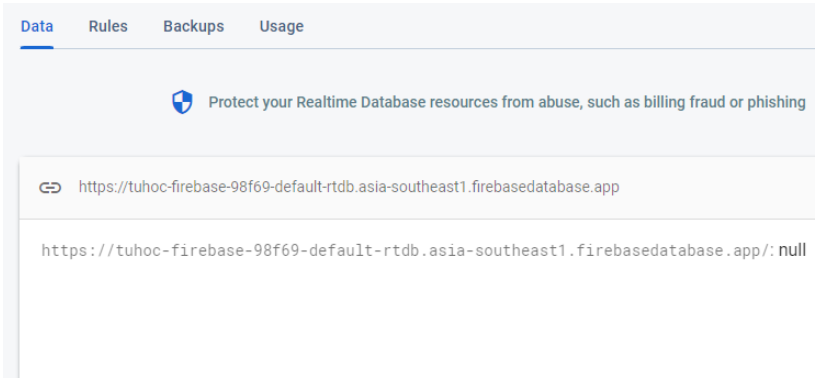
```
class InsertionActivity : AppCompatActivity() {

    private lateinit var dbRef : DatabaseReference

    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_insertion)

        dbRef = FirebaseDatabase.getInstance().getReference( path: "Employees")
        //xử lý sự kiện khi nhấn vào nút save
        btnSave.setOnClickListener { it: View!
            saveEmployeeData()
        }
    }
```

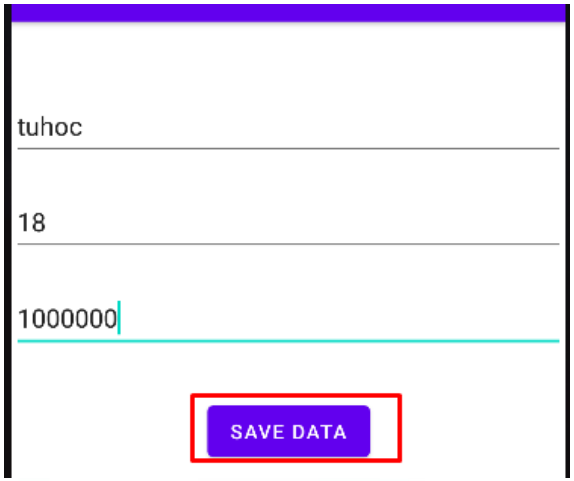
8. Code InsertionActivity xử lý thêm dữ liệu vào database :

✓ *Code hàm saveEmployeeData() : hàm xử lý thêm dữ liệu vào database*

```
private fun saveEmployeeData() {
    //getting value
    val empName = edtEmpName.text.toString()
    val empAge = edtEmpAge.text.toString()
    val empSalary = edtEmpSalary.text.toString()
    //kiểm tra xem các ô giá trị nhập có bị bỏ trống
    if (empName.isEmpty()) {
        edtEmpName.error = "Please enter name"
        return
    }
    if (empAge.isEmpty()) {
        edtEmpAge.error = "Please enter age"
        return
    }
    if (empSalary.isEmpty()) {
        edtEmpSalary.error = "Please enter salary"
        return
    }

    //đẩy dữ liệu
    val empId = dbRef.push().key!!
    val employee = EmployeeModel(empId,empName,empAge,empSalary)
    dbRef.child(empId).setValue(employee)
        .addOnCompleteListener {  it: Task<Void!>
            Toast.makeText( context: this, text: "Data insert thành công", Toast.LENGTH_SHORT).show()
        }.addOnFailureListener { err ->
            Toast.makeText( context: this, text: "Error ${err.message}", Toast.LENGTH_SHORT).show()
        }
}
```

❑ 9. Test insert data on this app ☺ :





❑ 10. Xoá data các ô nhập liệu sau khi insert thành công :

✓ *Code trong hàm saveEmployeeData() : hàm xử lý thêm dữ liệu vào database*

```
//đẩy dữ liệu
val empId = dbRef.push().key!!
val employee = EmployeeModel(empId,empName,empAge,empSalary)
dbRef.child(empId).setValue(employee)
    .addOnCompleteListener {  it: Task<Void!>
        Toast.makeText( context: this, text: "Data insert thành công", Toast.LENGTH_SHORT).show()
        //xoá trắng các ô nhập liệu
        edtEmpName.text.clear()
        edtEmpAge.text.clear()
        edtEmpSalary.text.clear()

    }.addOnFailureListener { err ->
        Toast.makeText( context: this, text: "Error ${err.message}", Toast.LENGTH_SHORT).show()
    }
```

YouTube Gà Lại Lập Trình

http://tuhoc.cc

tuhoc.cc

jacky chan

ly lien kiet

firebase kotlin part 2

View data on RecyclerView

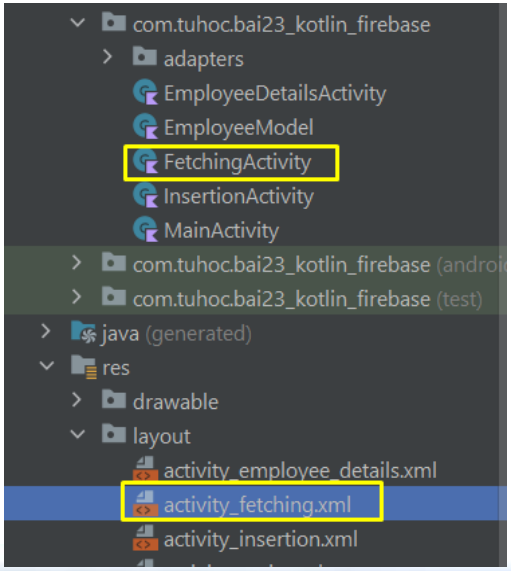
❑ 11. Hiển thị data lên RecyclerView :

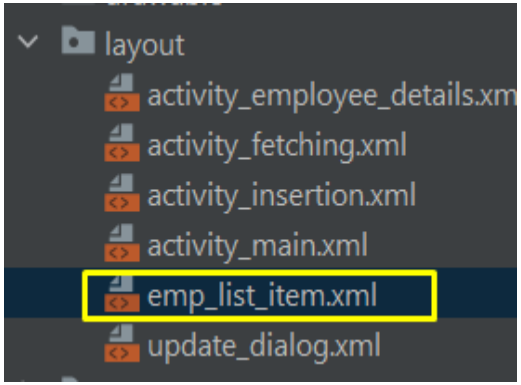
✓ *Các bước tiến hành*

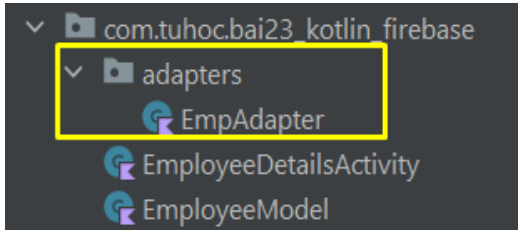
Step 1:
Tạo màn hình
FetchingActivity

Step 2:
Tạo layout mẫu 1 item cho recyclerView
emp_list_item.xml

Step 3:
Make Class
EmpAdapter

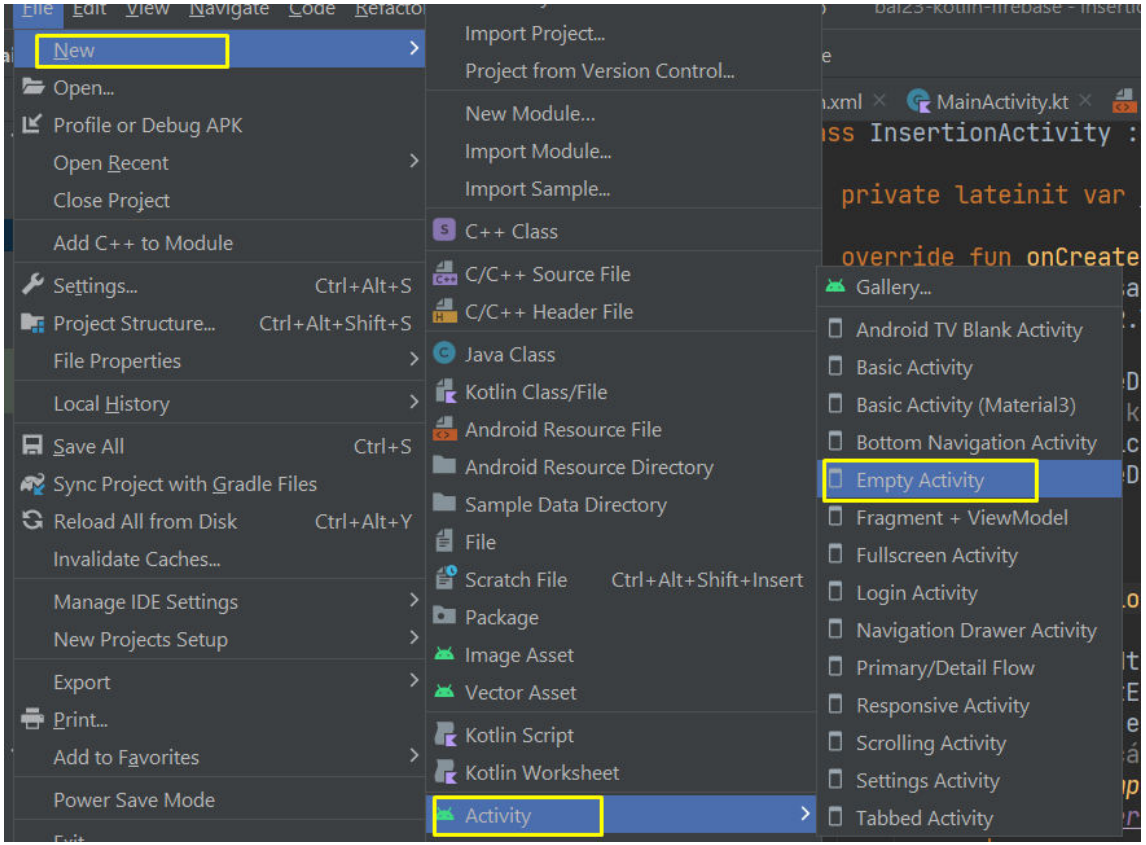


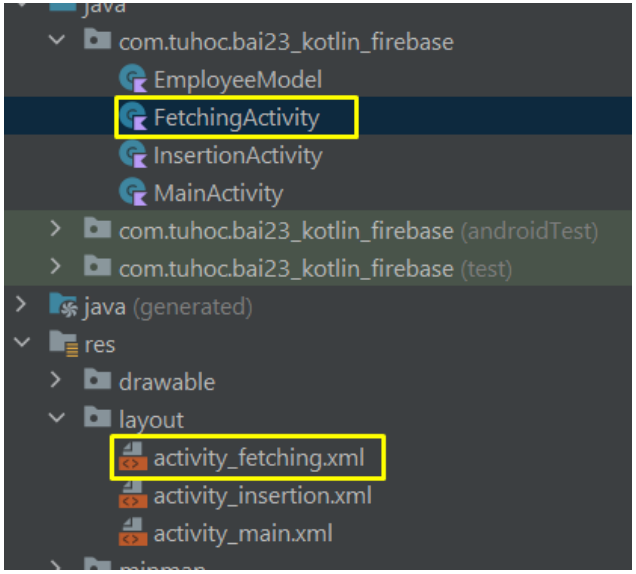




❑ 11. Hiển thị data lên RecyclerView :

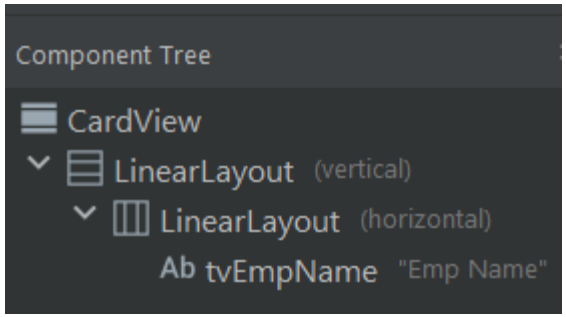
✓ *11-1 Step 1: Creat new Activity* FetchingActivity

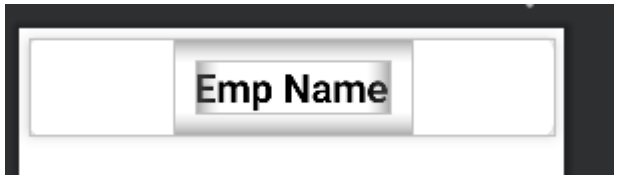




❑ 11. Hiển thị data lên RecyclerView :

✓ *11-2 Step2 Tạo layout cho 1 item* emp_list_item.xml

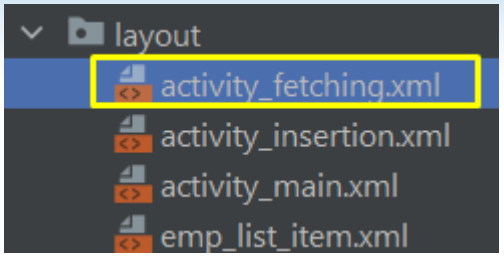


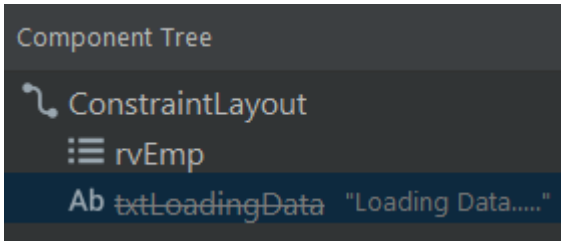


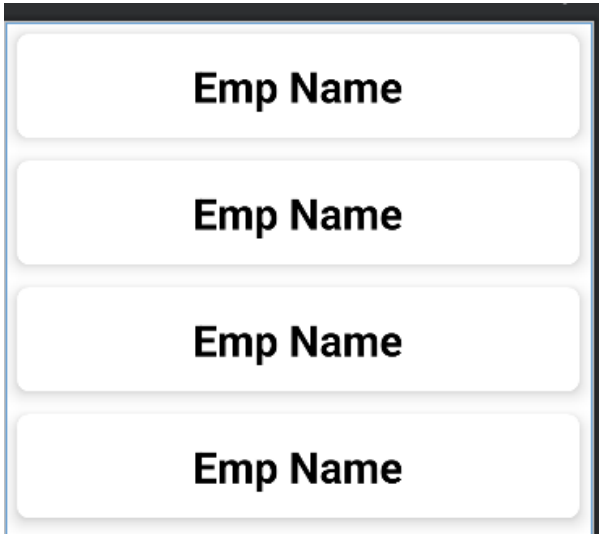

```xml
<?xml version="1.0" encoding="utf-8"?>
<androidx.cardview.widget.CardView xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    app:cardElevation="8dp"
    app:cardCornerRadius="8dp"
    android:layout_margin="8dp">

    <LinearLayout
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:gravity="center"
        android:padding="16dp"
        android:layout_gravity="center"
        android:orientation="vertical">

        <LinearLayout
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_gravity="center_vertical"
            android:orientation="horizontal">

            <TextView
                android:id="@+id/tvEmpName"
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:text="Emp Name"
                android:layout_gravity="center"
                android:gravity="center"
                android:textColor="@color/black"
                android:textSize="30sp"
                android:textStyle="bold"/>
        </LinearLayout>

    </LinearLayout>
```

❑ 11. Hiển thị data lên RecyclerView :

✓ *11-3 Code activity_fetching.xml*



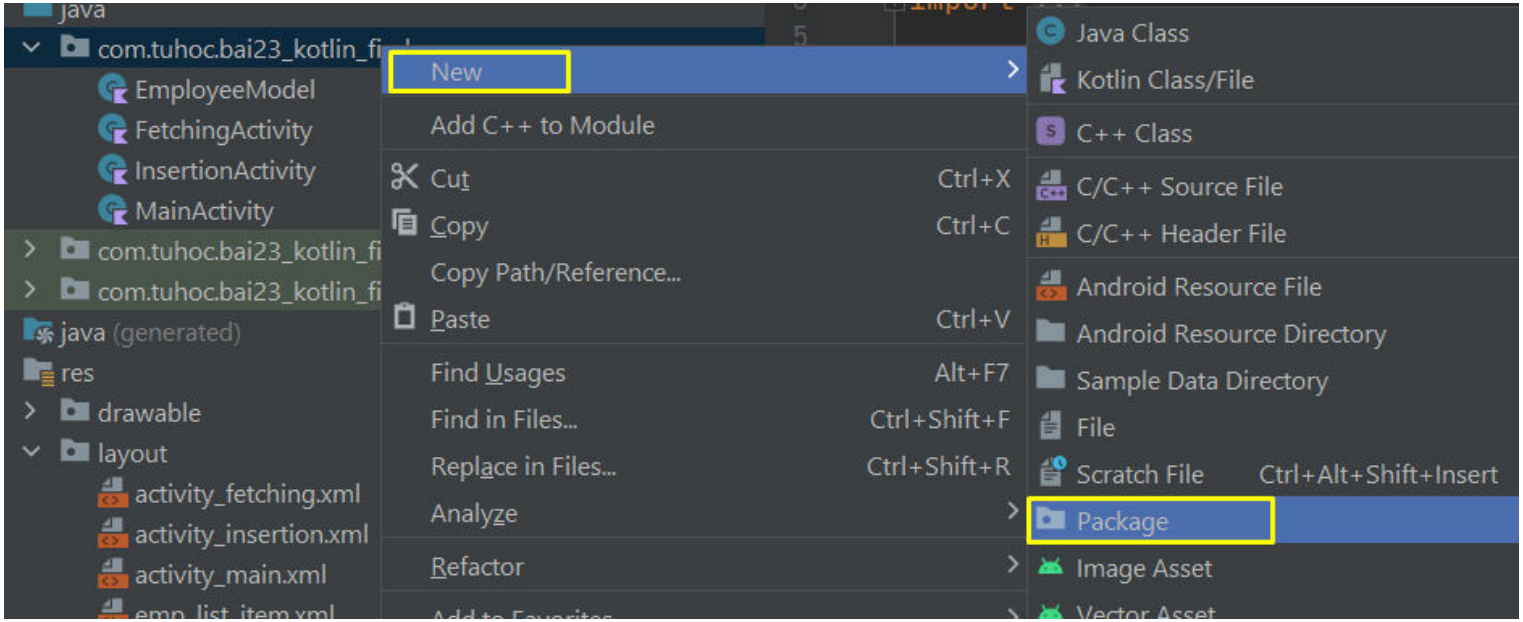


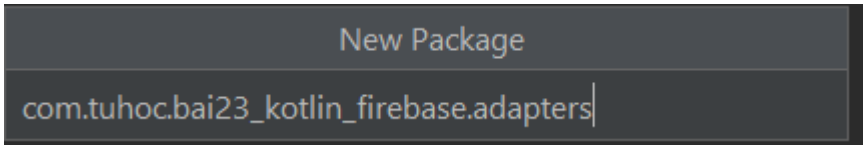


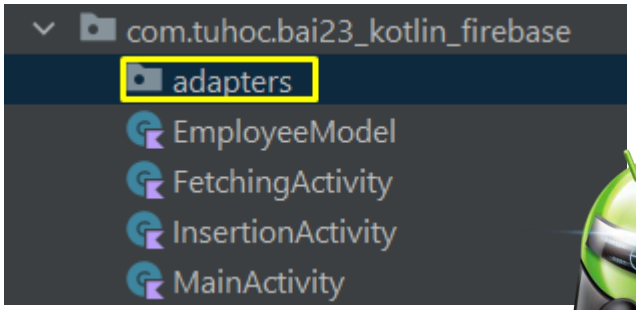

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout xmlns:androi
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".FetchingActivity">

    <androidx.recyclerview.widget.RecyclerView
        android:id="@+id/rvEmp"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:layout_marginStart="1dp"
        android:layout_marginTop="1dp"
        android:layout_marginEnd="1dp"
        android:layout_marginBottom="1dp"
        tools:listitem="@layout/emp_list_item"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent" />

    <TextView
        android:id="@+id/txtLoadingData"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Loading Data....."
        android:textSize="28sp"
        android:textColor="@color/black"
        android:textStyle="bold"
        android:visibility="gone"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent" />

</androidx.constraintlayout.widget.ConstraintLayout>
```

❑ 11. Hiển thị data lên gridview :

✓ *11-4 Step 3: Tạo adapter cho recyclerView*

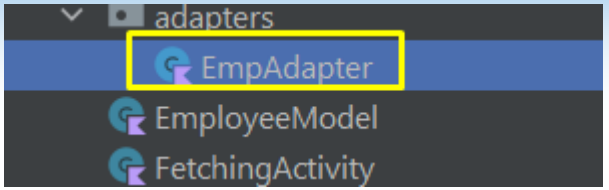






❑ 11. Hiển thị data lên RecyclerView :

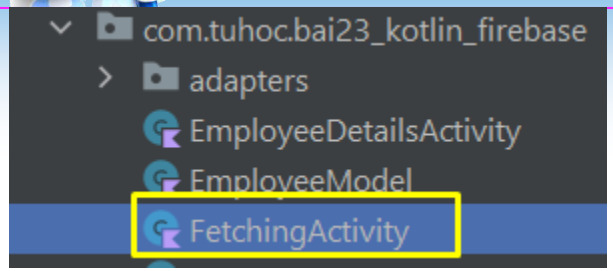
✓ *11-4 Step 3: Tạo adapter cho recyclerView*




```
package com.tuhoc.bai23_kotlin_firebase.adapter

import android.view.LayoutInflater
import android.view.View
import android.view.ViewGroup
import androidx.recyclerview.widget.RecyclerView
import com.tuhoc.bai23_kotlin_firebase.EmployeeModel
import com.tuhoc.bai23_kotlin_firebase.R
import kotlinx.android.synthetic.main.emp_list_item.view.*

//bài 16
class EmpAdapter(private val ds:ArrayList<EmployeeModel>) :RecyclerView.Adapter<EmpAdapter.ViewHolder>() {
    //tạo class viewholder
    class ViewHolder(itemView: View) : RecyclerView.ViewHolder(itemView)

    //ctrl + i
    override fun onCreateViewHolder(parent: ViewGroup, viewType: Int): ViewHolder {
        val itemView = LayoutInflater.from(parent.context)
            .inflate(R.layout.emp_list_item,parent, attachToRoot: false)
        return ViewHolder(itemView)
    }

    override fun onBindViewHolder(holder: ViewHolder, position: Int) {
        holder.itemView.apply { this: View
            tvEmpName.text = ds[position].empName
        }
    }

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

❑ 11. Hiển thị data lên RecyclerView :

✓ *11-5 Code FetchingActivity.kt*

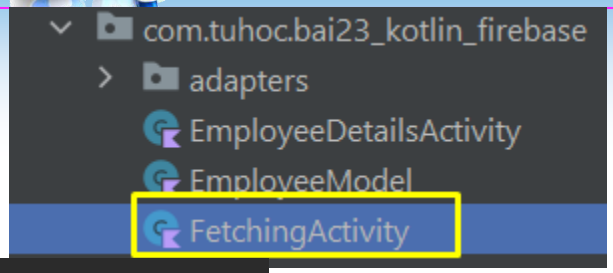



```
class FetchingActivity : AppCompatActivity() {

    private lateinit var ds:ArrayList<EmployeeModel>
    private lateinit var dbRef :DatabaseReference

    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_fetching)

        rvEmp.layoutManager = LinearLayoutManager( context: this)
        rvEmp.setHasFixedSize(true)
        ds = arrayListOf<EmployeeModel>()
        GetThongTinNV() // alt + enter
    }
```

❑ 11. Hiển thị data lên RecyclerView :

✓ *11-5 Code FetchingActivity.kt*

GetThongTinNV()

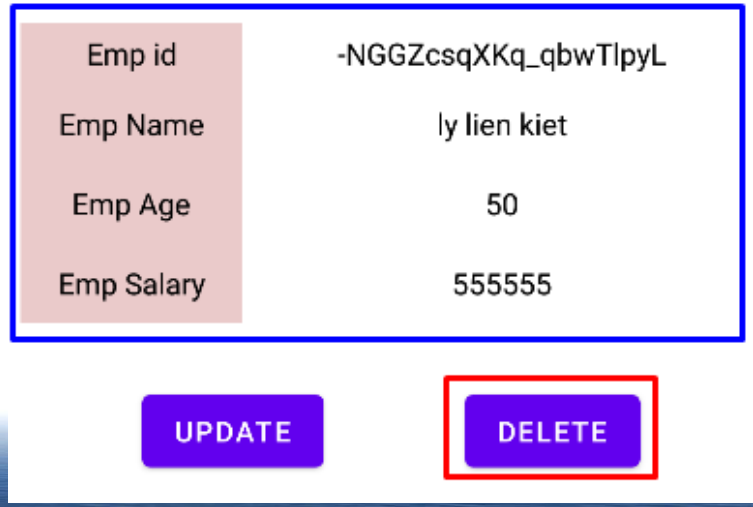



```
private fun GetThongTinNV() {
    rvEmp.visibility = View.GONE
    txtLoadingData.visibility = View.VISIBLE
    dbRef = FirebaseDatabase.getInstance().getReference( path: "Employees")
    //Để đọc dữ liệu tại một đường dẫn và lắng nghe các thay đổi,
    //hãy sử dụng addValueEventListener()
    //https://firebase.google.com/docs/database/android/read-and-write
    dbRef.addValueEventListener(object : ValueEventListener{
        //ctrl +i
        override fun onDataChange(snapshot: DataSnapshot) {
            ds.clear()
            if (snapshot.exists()){
                for(empSnap in snapshot.children){
                    val empData = empSnap.getValue(EmployeeModel::class.java)
                    ds.add(empData!!)
                }
                val mAdapter = EmpAdapter(ds)
                rvEmp.adapter = mAdapter
                rvEmp.visibility = View.VISIBLE
                txtLoadingData.visibility = View.GONE
            }
        }
        override fun onCancelled(error: DatabaseError) {
            TODO( reason: "Not yet implemented")
        }
    })
}
```

YouTube Gà Lại Lập Trình

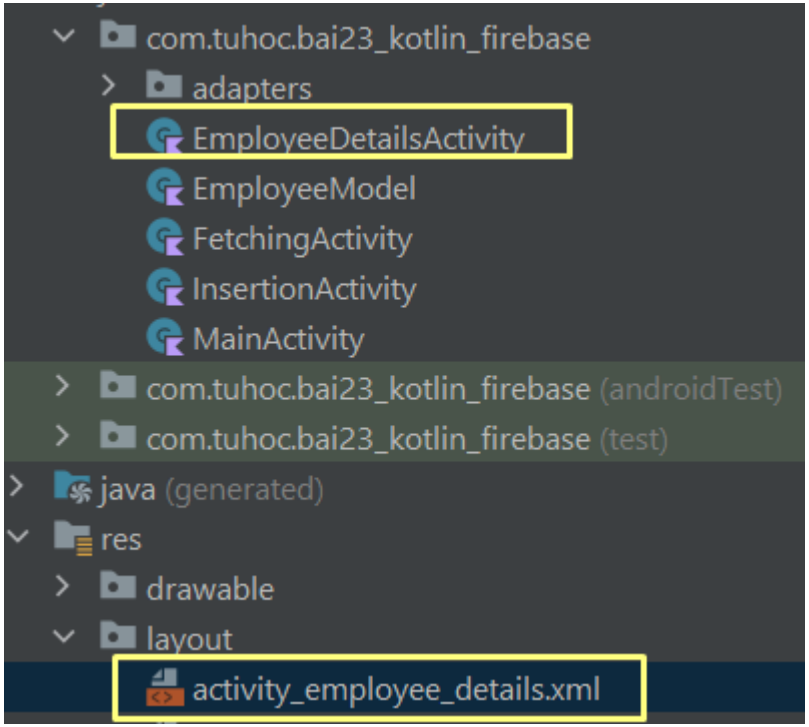
http://tuhoc.cc

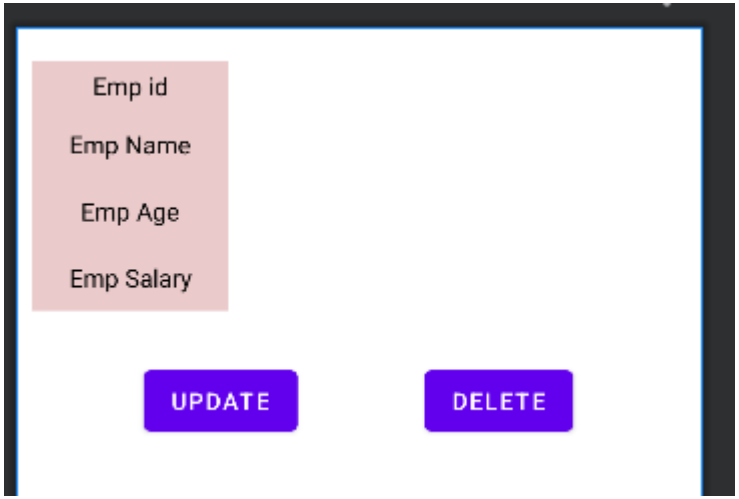
firebase kotlin part 3
View detail database
&
Delete Database



12. Hiển thị thông tin chi tiết data :

- ✓ *12-1 Tạo new Activity* ***EmployeeDetailsActivity***
- ✓ *Thiết kế giao diện* ***activity_employee_details.xml***





❑ 12. Hiển thị thông tin chi tiết data :

✓ *12-2 Code class adapter EmpAdapter*

```
//bai 16
class EmpAdapter(private val ds:ArrayList<EmployeeModel>) :RecyclerView.Adapter<EmpAdapter.ViewHolder>() {
    //code adapter lắng nghe sự kiện
    private lateinit var mListener: onItemClickListener
    interface onItemClickListener{
        fun onItemClick(position: Int)
    }
    fun setOnItemClickListener(clickListener: onItemClickListener){
        mListener = clickListener
    }

    //tạo class viewholder
    class ViewHolder(itemView: View,clickListener: onItemClickListener) : RecyclerView.ViewHolder(itemView){
        init {
            itemView.setOnClickListener { it: View!
                clickListener.onItemClick(adapterPosition)
            }
        }
    }

    //ctrl + i
    override fun onCreateViewHolder(parent: ViewGroup, viewType: Int): ViewHolder {
        val itemView = LayoutInflater.from(parent.context)
            .inflate(R.layout.emp_list_item,parent, attachToRoot: false)
        return ViewHolder(itemView,mListener)
    }
```

❑ 12. Hiển thị thông tin chi tiết data :

✓ *12-3 setOnItemClickListener RecyclerView FetchingActivity.kt*

FetchingActivity.kt

```
private fun GetThongTinNV() {
    rvEmp.visibility = View.GONE
    txtLoadingData.visibility = View.VISIBLE
    dbRef = FirebaseDatabase.getInstance().getReference( path: "Employees")
    //Để đọc dữ liệu tại một đường dẫn và lắng nghe các thay đổi,
    //hãy sử dụng addValueEventListener()
    //https://firebase.google.com/docs/database/android/read-and-write
    dbRef.addValueEventListener(object : ValueEventListener{
        //ctrl +i
        override fun onDataChange(snapshot: DataSnapshot) {
            ds.clear()
            if (snapshot.exists()){
                for(empSnap in snapshot.children){
                    val empData = empSnap.getValue(EmployeeModel::class.java)
                    ds.add(empData!!)
                }
                val mAdapter = EmpAdapter(ds)
                rvEmp.adapter = mAdapter
                //code lắng nghe sự kiện click lên item rv
                mAdapter.setOnItemClickListener(object : EmpAdapter.onItemClickListener{
                    //ctrl +i
                    override fun onItemClick(position: Int) {
                        val intent = Intent( packageContext: this@FetchingActivity,EmployeeDetailsActivity::class.java)
                        //put extras
                        intent.putExtra( name: "empId",ds[position].empId)
                        intent.putExtra( name: "empName",ds[position].empName)
                        intent.putExtra( name: "empAge",ds[position].empAge)
                        intent.putExtra( name: "empSalary",ds[position].empSalary)
                        startActivity(intent)
                    }
                })
```

❑ 12. Hiển thị thông tin chi tiết data :

✓ *12-4 Code EmployeeDetailsActivity nhận dữ liệu gửi từ FetchingActivity*

```
class EmployeeDetailsActivity : AppCompatActivity() {

    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_employee_details)

        setValuesToViews()

    }

    private fun setValuesToViews() {
        tvEmpId.text = intent.getStringExtra( name: "empId")
        tvEmpName.text = intent.getStringExtra( name: "empName")
        tvEmpAge.text = intent.getStringExtra( name: "empAge")
        tvEmpSalary.text = intent.getStringExtra( name: "empSalary")
    }

}
```

❑ 13. Xoá bản ghi :

EmployeeDetailsActivity.kt

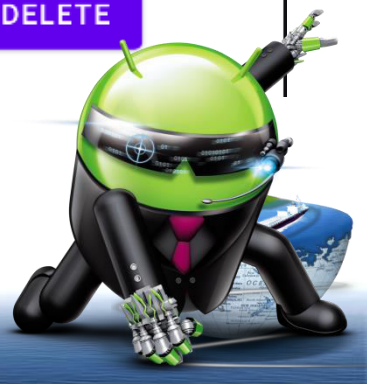
```
class EmployeeDetailsActivity : AppCompatActivity() {
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_employee_details)

        setValueToView()
        //code cho button delete
        btnDelete.setOnClickListener {  it: View!
            //alt + enter
            deleteRecord(
                intent.getStringExtra( name: "empId").toString()
            )
        }
    }

    private fun deleteRecord(id: String) {
        val dbRef = FirebaseDatabase.getInstance().getReference( path: "Employees").child(id)
        val mTask = dbRef.removeValue()
        mTask.addOnSuccessListener {  it: Void!
            Toast.makeText( context: this, text: "Employee data đã xoá",Toast.LENGTH_SHORT).show()
            val intent = Intent( packageContext: this,FetchingActivity::class.java)
            finish()
            startActivity(intent)
        }.addOnFailureListener { err->
            Toast.makeText( context: this, text: "Delete err ${err.message}",Toast.LENGTH_SHORT).show()
        }

    }
```

YouTube Gà Lại Lập Trình

http://tuhoc.cc

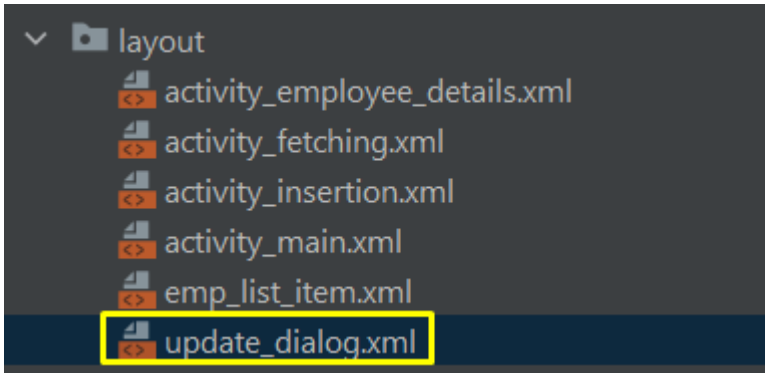
firebase kotlin part 4

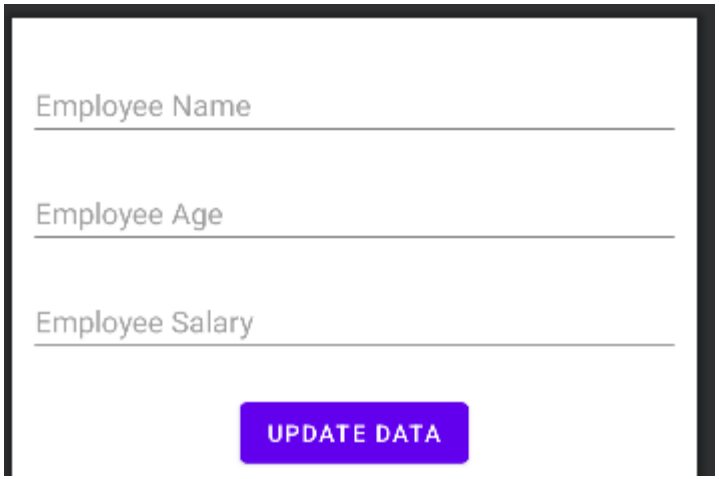
Update Database

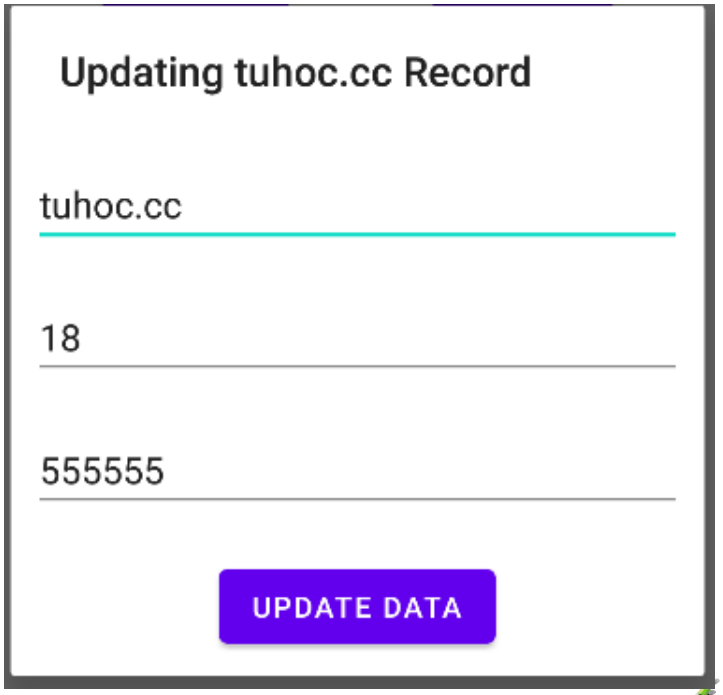


❑ 14. Update data :

✓ *14-1 Thiết kế giao diện update_dialog.xml*







❑ 14. Update data :

✓ *14-2 Lắng nghe sự kiện click lên btnUpdate*

EmployeeDetailsActivity.kt

```
class EmployeeDetailsActivity : AppCompatActivity() {

    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_employee_details)

        setValuesToViews()

        btnUpdate.setOnClickListener {  it: View!
            openUpdateDialog(
                intent.getStringExtra( name: "empId").toString(),
                intent.getStringExtra( name: "empName").toString()
            )
        }
```

❑ 14. Update data :

- ✓ *14-3 code hàm openUpdateDialog()*
- ✓ *Hiển thị thông tin lên Dialog*

```
private fun openUpdateDialog(
    empId: String,
    empName: String
) {
    val mDialog = AlertDialog.Builder( context: this)
    val inflater = layoutInflater
    val mDialogView = inflater.inflate(R.layout.update_dialog, root: null)
    mDialog.setView(mDialogView)

    val etEmpName = mDialogView.findViewById<EditText>(R.id.etEmpName)
    val etEmpAge = mDialogView.findViewById<EditText>(R.id.etEmpAge)
    val etEmpSalary = mDialogView.findViewById<EditText>(R.id.etEmpSalary)
    val btnUpdateData = mDialogView.findViewById<Button>(R.id.btnUpdateData)

    etEmpName.setText(intent.getStringExtra( name: "empName").toString())
    etEmpAge.setText(intent.getStringExtra( name: "empAge").toString())
    etEmpSalary.setText(intent.getStringExtra( name: "empSalary").toString())

    mDialog.setTitle("Updating $empName Record")

    val alertDialog = mDialog.create()
    alertDialog.show()

}
```

❑ 14. Update data :

- ✓ *14-4 code hàm openUpdateDialog()*
- ✓ *Update data khi click button btnUpdateData*

```
    btnUpdateData.setOnClickListener { it: View!
        updateEmpData(
            empId,
            etEmpName.text.toString(),
            etEmpAge.text.toString(),
            etEmpSalary.text.toString()
        )

        Toast.makeText(applicationContext, text: "Employee Data Updated", Toast.LENGTH_LONG).show()

        //we are setting updated data to our textviews
        tvEmpName.text = etEmpName.text.toString()
        tvEmpAge.text = etEmpAge.text.toString()
        tvEmpSalary.text = etEmpSalary.text.toString()

        alertDialog.dismiss()
    }
}
private fun updateEmpData(
    id: String,
    name: String,
    age: String,
    salary: String
) {
    val dbRef = FirebaseDatabase.getInstance().getReference( path: "Employees").child(id)
    val empInfo = EmployeeModel(id, name, age, salary)
    dbRef.setValue(empInfo)
}
```