

We can use Android Data Binding to bind click and other user interface events.

To do that we need to use an inner class within the corresponding activity or fragment.

We need to create a class (inner class) with methods for different click events and declare a variable in the XML layout file to access that class.

1) Create an inner class for click handlers.

```
public class MainActivityClickHandlers {
    Context context;

    public MainActivityClickHandlers(Context context) {
        this.context = context;
    }

    public void onEnrollbuttonClicked(View view) {
        Toast.makeText(context, text: "Enroll Button clicked!", Toast.LENGTH_LONG).show();
    }

    public void onCancelbuttonClicked(View view) {
        Toast.makeText(context, text: "Cancel Button clicked!", Toast.LENGTH_LONG).show();
    }
}
```

2) Declare a reference variable for the click handlers, in the XML file.

```

<?xml version="1.0" encoding="utf-8"?>
<layout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto">

    <data>
        <variable
            name="student"
            type="com.anushka.androidtutz.bindingdemo1.Student"/>

        <variable
            name="clickHandler"
            type="com.anushka.androidtutz.bindingdemo1.MainActivity.MainActivityClickHandlers"/>
    </data>

    <android.support.constraint.ConstraintLayout
        xmlns:tools="http://schemas.android.com/tools"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        tools:context=".MainActivity">

```

- 3) Construct the inner class (handler class) passing the context(in the onCreate method) and set the handler class.

```

protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);

    activityMainBinding=DataBindingUtil.setContentView( activity: this,R.layout.activity_main);
    activityMainBinding.setStudent(getCurrentStudent());

    handlers=new MainActivityClickHandlers( context: this);
    activityMainBinding.setClickHandler(handlers);
}

```

- 4) Set a call to a method of the inner class as the value of the onClick property(in XML Layout).

```
<Button
    android:id="@+id/button"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginStart="8dp"
    android:layout_marginLeft="8dp"
    android:layout_marginTop="8dp"
    android:layout_marginEnd="8dp"
    android:layout_marginRight="8dp"
    android:layout_marginBottom="8dp"
    android:text="Enroll"
    android:onClick="@{clickHandler::onEnrollbuttonClicked}"
    android:textSize="36sp"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/tvStudentEmail"
    app:layout_constraintVertical_bias="0.211" />
```

Thank you for reading.

If you have any question please ask from me (using Q & A or Messages).

I am always happy to support you.

Happy Learning.....

Anushka Madusanka

