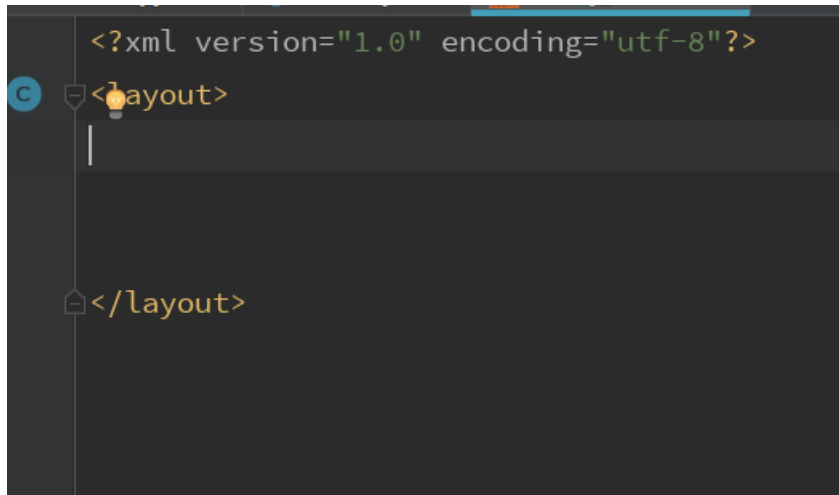


1) To tell Android data binding library that this layout is going to use data binding, we need to include these content between **layout tags**.



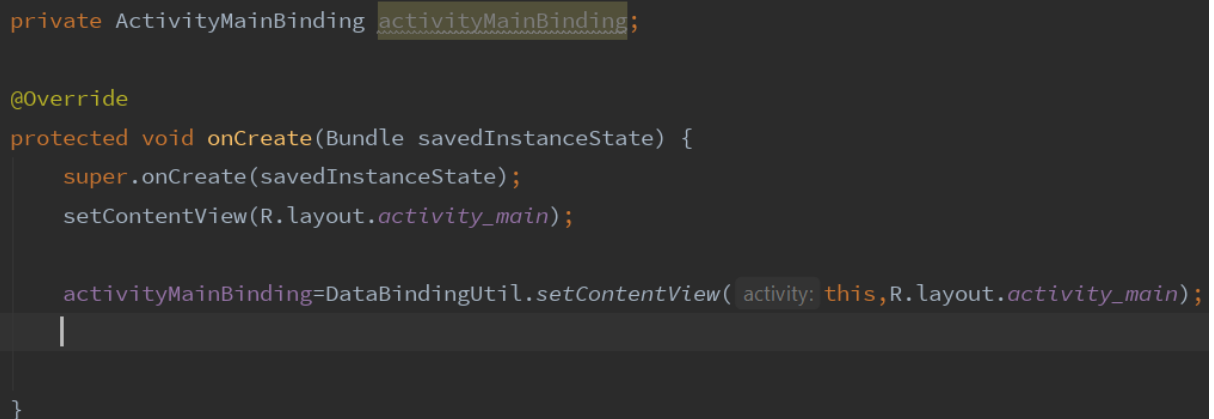
```
<?xml version="1.0" encoding="utf-8"?>
<layout>
|
</layout>
```

2) When we rebuild the app, data binding library will recognize the layout as a layout for data binding and it will **generate a corresponding class** for it.

The code generator creates the name using the name of the xml file.

(If the XML layout file's name is `activity_main`, name of the newly generated class will be `ActivityMainBinding`)

3) In the Activity we can create an instance of newly generated class.



```
private ActivityMainBinding activityMainBinding;

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

    activityMainBinding=DataBindingUtil.setContentView( activity: this,R.layout.activity_main);
|
}
```

4) Now we need to pass objects to the layout. To do that we need to create reference variables in the layout. We can do this inside “data” tags.

```
<?xml version="1.0" encoding="utf-8"?>
<layout>

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

    <android.support.constraint.ConstraintLayout
        xmlns:android="http://schemas.android.com/apk/res/android"
        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"
```

5) Data binding library will generate setter methods according to the variable name we have given, in above case, it will generate a “setStudent(Student student)” method.

We can pass our instances to the setter methods.

```
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());  
}  
  
private Student getCurrentStudent(){  
  
    Student student = new Student();  
    student.setStudentName("Alex");  
    student.setStudentEmail("alex@gmail.com");  
    return student;  
}
```

]

6) We can use properties of the passed object within the layout file. Using “@{ }”.

```
<TextView
    android:id="@+id/tvStudentEmail"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="@{student.studentEmail}"
    android:textSize="36sp"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintHorizontal_bias="0.465"
    app:layout_constraintLeft_toLeftOf="parent"
    app:layout_constraintRight_toRightOf="parent"
    app:layout_constraintTop_toTopOf="parent"
    app:layout_constraintVertical_bias="0.384" />
```

```
<TextView
    android:id="@+id/tvStudentName"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="@{student.studentName}"
    android:textSize="36sp"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintHorizontal_bias="0.464"
    app:layout_constraintLeft_toLeftOf="parent"
    app:layout_constraintRight_toRightOf="parent"
    app:layout_constraintTop_toTopOf="parent"
```

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

