

Session 2 - Exercises

V1.0

JUAN RONDON

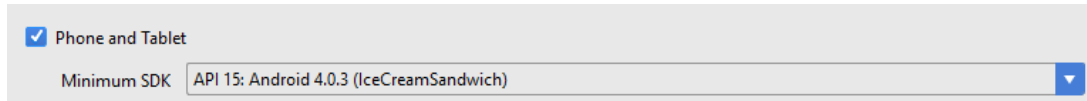


Table of Contents

Part 1 - Listeners	2
Exercise 1 - Getting familiar with some of the View Widgets	2
Part 2	7
Android Note taking application	7

Part 1 - Listeners

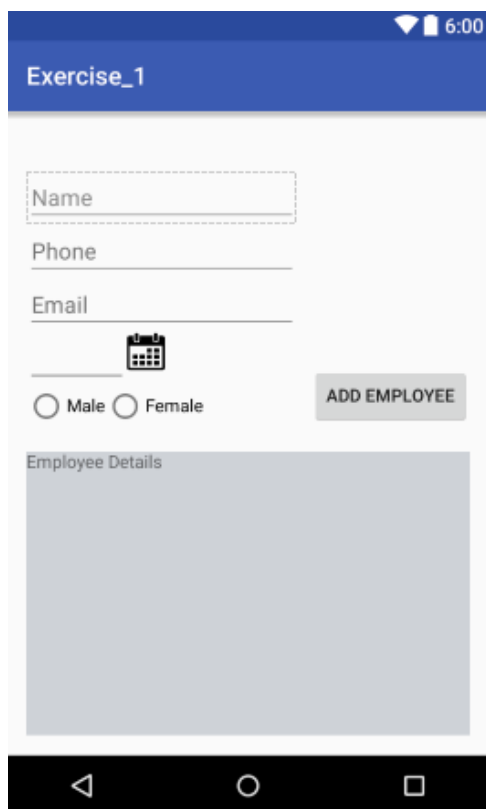
Create a new project for the following exercise in Android Studio. In the form factor selection make sure to select:



Exercise 1 - Getting familiar with some of the View Widgets

For all the widgets set the hint property i.e. name, Phone, etc.

- Create an Activity with the following View widgets:
 - EditText** for the name
 - EditText** for the phone number
 - EditText** for the Email
 - EditText** for the date of birth (set property **android:focusable="false"**)
 - ImageView** for the calendar image (give **width** and **length** of **30dp**)
 - RadioGroup** with two inner **RadioButton** widgets for the gender. (make the **orientation** of the **RadioGroup** = **horizontal**)
 - Button** with text set to **"Add"**
 - ScrollView** with an inner **TextView** used to display all the information entered when the Add Employee button is pressed.



Find the layout code in the next page

```

<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:paddingBottom="@dimen/activity_vertical_margin"
    android:paddingLeft="@dimen/activity_horizontal_margin"
    android:paddingRight="@dimen/activity_horizontal_margin"
    android:paddingTop="@dimen/activity_vertical_margin"
    tools:context="androidcourse.exercise_1.MainActivity">

    <EditText
        android:id="@+id/name"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignParentLeft="true"
        android:layout_alignParentStart="true"
        android:layout_alignParentTop="true"
        android:layout_marginTop="32dp"
        android:ems="10"
        android:hint="Name"
        android:inputType="textPersonName" />

    <EditText
        android:id="@+id/phone"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignParentLeft="true"
        android:layout_alignParentStart="true"
        android:layout_below="@+id/name"
        android:ems="10"
        android:hint="Phone"
        android:inputType="phone" />

    <EditText
        android:id="@+id/email"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignParentLeft="true"
        android:layout_alignParentStart="true"
        android:layout_below="@+id/phone"
        android:ems="10"
        android:hint="Email"
        android:inputType="textEmailAddress" />

    <EditText
        android:id="@+id/dob"
        android:layout_width="110dp"
        android:layout_height="wrap_content"
        android:layout_alignParentLeft="true"
        android:layout_alignParentStart="true"
        android:layout_below="@+id/email"
        android:focusable="false" />

    <ImageView
        android:id="@+id/calendar"
        android:layout_width="30dp"
        android:layout_height="30dp"
        android:layout_below="@+id/email"
        android:layout_toEndOf="@+id/dob"
        android:layout_toRightOf="@+id/dob"
        android:src="@drawable/calendar" />

```

```

<RadioGroup
    android:id="@+id/radioGroup"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_alignEnd="@+id/email"
    android:layout_alignParentLeft="true"
    android:layout_alignParentStart="true"
    android:layout_alignRight="@+id/email"
    android:layout_below="@+id/dob"
    android:orientation="horizontal">

    <RadioButton
        android:id="@+id/male"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignEnd="@+id/date"
        android:layout_alignRight="@+id/date"
        android:layout_centerVertical="true"
        android:checked="false"
        android:text="Male" />

    <RadioButton
        android:id="@+id/female"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_centerVertical="true"
        android:layout_toEndOf="@+id/male"
        android:layout_toRightOf="@+id/male"
        android:checked="false"
        android:text="Female" />

</RadioGroup>

<ScrollView
    android:layout_width="wrap_content"
    android:id="@+id/scroll"
    android:layout_alignParentBottom="true"
    android:layout_alignParentEnd="true"
    android:layout_alignParentLeft="true"
    android:fadeScrollbars="false"
    android:layout_alignParentRight="true"
    android:layout_alignParentStart="true"
    android:layout_below="@+id/radioGroup"
    android:layout_marginTop="20dp"
    android:layout_height="wrap_content">

    <TextView
        android:id="@+id/details"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:background="#4567"
        android:gravity="top"
        android:hint="Employee Details" />
</ScrollView>

<Button
    android:id="@+id/addId"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_above="@+id/scroll"
    android:layout_alignEnd="@+id/scroll"
    android:layout_alignRight="@+id/scroll"
    android:text="Add Employee" />
</RelativeLayout>

```

2. Now you will be creating a folder inside **androidcourse.exercise_1** package (folder) called **Models**. Right click in **androidcourse.exercise_1** and select new package. Give it a name of **Models**.
3. Create a java class called **Employee** inside **Models** folder with the following properties:
 - id, name, phone number, email, dob as Date, gender.
 - Create a constructor to initialize all the fields.
 - Create a method to return all the details of the Employee object.
i.e.
id: 1
Name: Test name
Phone: 9545432
Email: test@test.com
DOB: 23/03/1982
Gender: Male

Employee code:

```
package androidcourse.exercise_1.Models;

import java.text.ParseException;
import java.text.SimpleDateFormat;
import java.util.Date;

public class Employee {
    private SimpleDateFormat sdf = new SimpleDateFormat("dd/mm/yyyy");
    private static int NEXT_ID = 1;
    private int id;
    private String name;
    private String phone;
    private String email;
    private Date dob;
    private String gender;

    public Employee(String name, String phone, String email, String dob,
String gender) {
        this.id = NEXT_ID;
        NEXT_ID++;
        this.name = name;
        this.phone = phone;
        this.email = email;
        try {
            this.dob = sdf.parse(dob);
        } catch (ParseException error) {
        }
        this.gender = gender;
    }

    public String getDetails() {
        String details = "Id: " + id;
        details += "\nName: " + name;
        details += "\nPhone: " + phone;
        details += "\nEmail: " + email;
        details += "\nDate Of Birth: " + sdf.format(dob);
        details += "\nGender: " + gender + "\n\n";
        return details;
    }
}
```

- Once the model class is completed we are ready to create all the listeners for our Activity (one listener for the Add Employee button and another for the Date of birth (calendar image)).
- Open **MainActivity.java** class and copy the required code that is missing.
Pay attention at all the required **Import** statements.

```
package androidcourse.exercise_1;

//Import statements omitted for convenience

public class MainActivity extends AppCompatActivity {

    private EditText name;
    private EditText phone;
    private EditText email;
    private EditText dob;
    private RadioGroup gender;
    private RadioButton genderSex;
    private TextView details;
    private int mYear, mMonth, mDay;

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

        name = (EditText) findViewById(R.id.name);
        phone = (EditText) findViewById(R.id.phone);
        email = (EditText) findViewById(R.id.email);
        dob = (EditText) findViewById(R.id.dob);
        gender = (RadioGroup) findViewById(R.id.radioGroup);
        Button add = (Button) findViewById(R.id.addId);
        details = (TextView) findViewById(R.id.details);

        ImageView date = (ImageView) findViewById(R.id.calendar);

        //listener for calendar image
        date.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                // Process to get Current Date
                final Calendar c = Calendar.getInstance();
                mYear = c.get(Calendar.YEAR);
                mMonth = c.get(Calendar.MONTH);
                mDay = c.get(Calendar.DAY_OF_MONTH);

                DatePickerDialog.OnDateSetListener listener = new DatePickerDialog.OnDateSetListener() {
                    @Override
                    public void onDateSet(DatePicker view, int year, int monthOfYear, int dayOfMonth) {
                        dob.setText(dayOfMonth + "/" + (monthOfYear + 1) + "/" + year);
                    }
                };
                DatePickerDialog dp = new DatePickerDialog(MainActivity.this, listener, mYear, mMonth, mDay);
                dp.show();
            }
        });

        //listener for Add button
        add.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                String empName = name.getText().toString();
                String empPhone = phone.getText().toString();
                String empEmail = email.getText().toString();
                String empDob = dob.getText().toString();
                int selectedId = gender.getCheckedRadioButtonId();

                // find the RadioButton by id
                genderSex = (RadioButton) findViewById(selectedId);
                String empGender = genderSex.getText().toString();

                Employee emp = new Employee(empName, empPhone, empEmail, empDob, empGender);
                String text = emp.getDetails();
                details.setText(details.getText() + text);
            }
        });
    }
}
```

- Open your application in the emulator/device and test the functionality of both listeners.

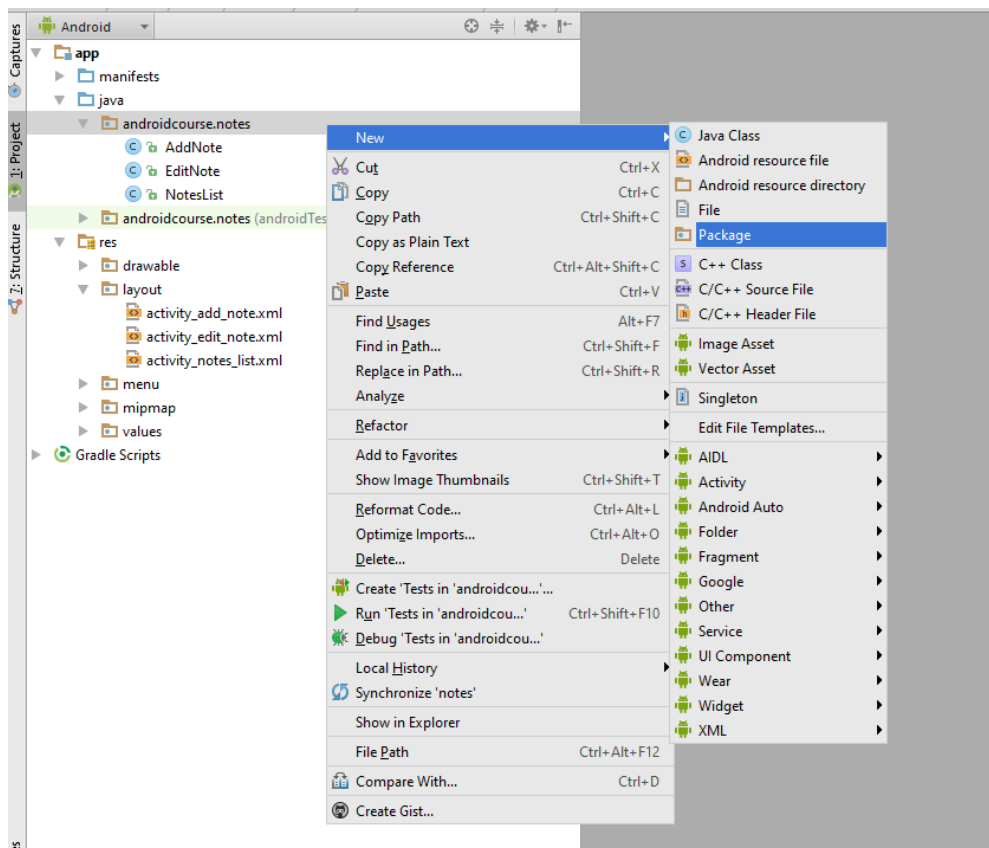
Part 2

Android Note taking application

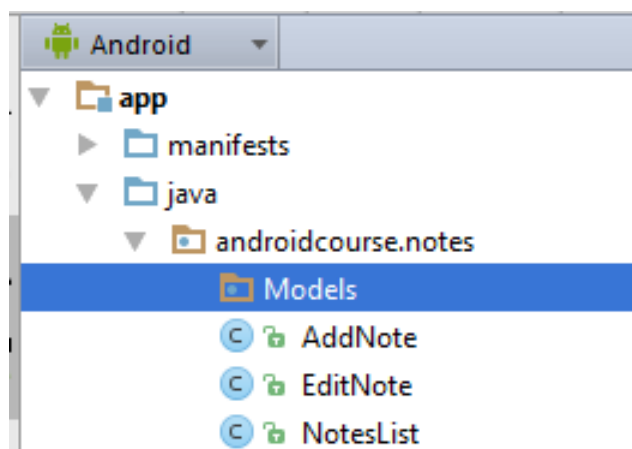
You will be adding the brains of the application (note class).

Open your notes app solution from Session 1 (Located in GitHub).

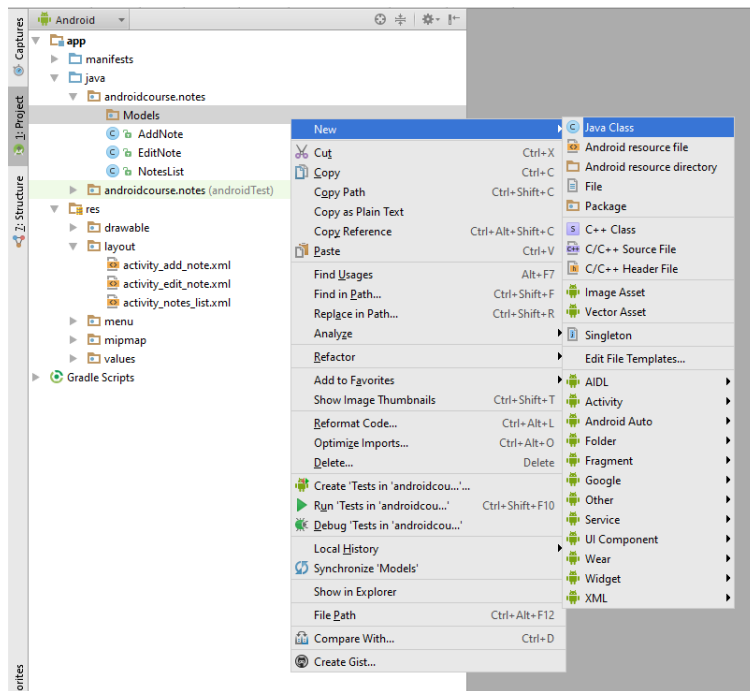
1. Create a new package inside **androidcourse.notes** package and name it **Models**.



2. Once the package is created your project structure should reflect the following image:



3. Create a java class named **Note** inside **Models** package.



4. After creating the Note class, you will be adding the following private instance variables to it:
- **id** as int
 - **title** as String
 - **content** as String
 - **lastModified** as Date
 - **password** as String
 - **static int NEXT_ID = 0;** This static property will be used to initialize **id** property for each of the note objects.

```
private int id;
private String title;
private String content;
private Date lastModified;
private String password;
private static int NEXT_ID = 0;
```

5. Next you will be creating **two** constructors for the class, the first one will be used to create notes with no password and the second one will be used in order to create notes that contain a password.

```
public Note(String title, String content) {
    id = NEXT_ID++;
    this.title = title;
    this.content = content;
    lastModified = new Date();
}

public Note(String title, String content, String password) {
    id = NEXT_ID++;
    this.title = title;
    this.content = content;
    this.password = password;
    lastModified = new Date();
}
```

6. Your class should look as follows:

```

package androidcourse.notes.Models;

import java.util.Date;

public class Note {
    private int id;
    private String title;
    private String content;
    private Date lastModified;
    private String password;
    private static int NEXT_ID = 1;

    public Note(String title, String content) {
        id = NEXT_ID++;
        this.title = title;
        this.content = content;
        lastModified = new Date();
    }

    public Note(String title, String content, String password) {
        id = NEXT_ID++;
        this.title = title;
        this.content = content;
        this.password = password;
        lastModified = new Date();
    }
}

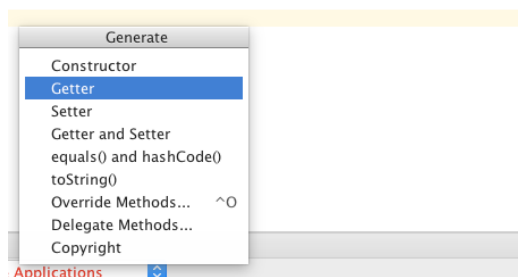
```

7. Now create methods to get the values for all the instance variables. (to do so under the constructors but inside the class press **ALT + Insert** and select getter.

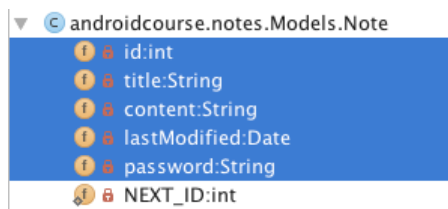
```

public Note(String title, String content, String password) {
    id = NEXT_ID++;
    this.title = title;
    this.content = content;
    this.password = password;
    lastModified = new Date();
}

```



8. From the window highlight all the properties except from Next_ID and Android Studio will create all the get methods for you! Once you are done click OK button.



9. We need an extra method in order to display the last modified date in an appropriate format for the user. Create the following method after the last generated get method.

```

public String dateFormatted() {
    SimpleDateFormat sdf = new SimpleDateFormat("MMM dd - HH:mm");
    return "Last edited on: " + sdf.format(lastModified);
}

```

Notice the red colour for **SimpleDateFormat** class and **format** method? You will need to import **SimpleDateFormat** class because it lives in a different package than our Note class.

First move the mouse on top of **SimpleDateFormat** and pres **Alt+Enter** in order to import the required classes.

SimpleDateFormat is a class that helps us to format dates in java. In this case we want our date to be displayed as “Month name abbreviated (MMM) followed by day with two digits (dd) followed by – followed by hour and finally minutes.”

10. Once you import the **SimpleDateFormat** class, the completed **Note** class should look like the following code:

```

public class Note {
    private int id;
    private String title;
    private String content;
    private Date lastModified;
    private String password;
    private static int NEXT_ID = 1;

    public Note(String title, String content) {
        id = NEXT_ID++;
        this.title = title;
        this.content = content;
        lastModified = new Date();
    }

    public Note(String title, String content, String password) {
        id = NEXT_ID++;
        this.title = title;
        this.content = content;
        this.password = password;
        lastModified = new Date();
    }

    public int getId() {
        return id;
    }

    public String getTitle() {
        return title;
    }

    public String getContent() {
        return content;
    }

    public Date getLastModified() {
        return lastModified;
    }

    public String getPassword() {
        return password;
    }

    public String dateFormatted() {
        SimpleDateFormat sdf = new SimpleDateFormat("MMM dd - HH:mm");
        return "Last edited on: " + sdf.format(lastModified);
    }
}

```

Now you will be creating references for all the literal strings used in the app so far inside String.xml resource file.