Android UI Control [EditText]

Android – UI Control

EditText

- **EditText** is a user interface control which is used to allow the user to enter or modify the text.
- While using **EditText** control in our android applications, we need to specify the type of data the text field can accept using the **inputType** attribute.
- For example, if it accept plain text, then we need to specify the inputType as "**text**". In case if **EditText** field is for password, then we need to specify the inputType as "**textPassword**".
- In android, we can create **EditText** control in two ways either in XML layout file or create it in Activity file programmatically.

Android – UI Control

EditText

Input Type Formats

An EditText box could be set to accept input strings satisfying a particular pattern such as: numbers (with and without decimals or sign), phones, dates, times, uris, etc.

Setting the EditText box to accept a particular choice of data-type, is done through the XML clause

```
android:inputType="choices"
```

where **choices** include any of the single values shown in the figure. You may combine types, for instance: `textCapWords|textAutoCorrect`
Accepts text that capitalizes every word, incorrect words are automatically changed (for instance 'teh' is converted into 'the', and so on.

Android – UI Control

Create a EditText in Layout File

```
<?xml version="1.0" encoding="utf-8"?>
    <LinearLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
       android:layout_width="match_parent"
       android:layout_height="match_parent"
       android:orientation="vertical" >
       <EditText
          android:id="@+id/txtSub"
          android:layout_width="match_parent"
          android:layout_height="wrap_content"
          android:hint="Subject"
          android:inputType="text"/>
    </LinearLayout>
```

Android – UI Control

Create EditText Control in Activity File

```
public class MainActivity extends AppCompatActivity {
        @Override
        protected void onCreate(Bundle savedInstanceState) {
            super.onCreate(savedInstanceState);
            setContentView(R.layout.activity_main);
            LinearLayout linearLayout =  (LinearLayout)
    findViewById(R.id.linearlayout);
            EditText et = new EditText(this);
            et.setHint("Subject");
            linearLayout.addView(et);
        }
    }
```


Android – UI Control

Set the Text of Android EditText

- We can set the text of **EditText** control either while declaring it in **Layout** file or by using **setText()** method in Activity file.

- Following is the example to set the text of TextView control while declaring it in XML Layout file.

```
<EditText
   android:id="@+id/editText1"
   android:layout_width="match_parent"
   android:layout_height="wrap_content"
   android:text="Welcome to DDU" />
```

- Following is another way to set the text of EditText control programmatically in activity file using setText() method.

```
EditText et = (EditText)findViewById(R.id.editText1);
    et.setText("Welcome to DDU");
```

Android – UI Control

Get Text of Android EditText

- We can get the text of **EditText** control by using **getText()** method in Activity file.

- Following is the example to get text of **EditText** control programmatically in activity file using **getText()** method.

```
public class MainActivity extends AppCompatActivity {
   @Override
   protected void onCreate(Bundle savedInstanceState) {
      super.onCreate(savedInstanceState);
      setContentView(R.layout.activity_main);
      EditText et = (EditText) findViewById(R.id.txtSub);
      String name = et.getText().toString();
   }
}
```


Android – UI Control

Android EditText Attributes

Attribute	Description
android:id	It is used to uniquely identify the control
android:gravity	It is used to specify how to align the text like left, right, center, top, etc.
android:text	It is used to set the text.
android:hint	It is used to display the hint text when text is empty
android:textColor	It is used to change the color of the text.
android:textColorHint	It is used to change the text color of hint text.
android:textSize	It is used to specify the size of the text.

Android – UI Control

Android EditText Attributes

Attribute	Description
android:textStyle	It is used to change the style (bold, italic, bolditalic) of text.
android:background	It is used to set the background color for edit text control
android:ems	It is used to make the textview be exactly this many ems wide.
android:width	It makes the TextView be exactly this many pixels wide.
android:height	It makes the TextView be exactly this many pixels tall.
android:maxWidth	It is used to make the TextView be at most this many pixels wide.

Android – UI Control

Android EditText Attributes

Attribute	Description
android:minWidth	It is used to make the TextView be at least this many pixels wide.
android:textAllCaps	It is used to present the text in all CAPS
android:typeface	It is used to specify the Typeface (normal, sans, serif, monospace) for the text.
android:textColorHighlight	It is used to change the color of text selection highlight.
android:inputType	It is used to specify the type of text being placed in text fields.
android:fontFamily	It is used to specify the fontFamily for the text.
android:editable	If we set false, EditText won't allow us to enter or modify the text

Android – UI Control

Android EditText Example

- In this example we will be using multiple **EditText** controls with different input types like password, phone, etc. in LinearLayout.
- Create a new android application using android studio and give names as **EditTextExample**.
- Now open an **activity_main.xml** file from **\res\layout** path and write the code like as shown below

Android – UI Control

Android EditText Example

activity_main.xml

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
xmlns:android="http://schemas.android.com/apk
/res/android"
   android:layout_width="match_parent"
   android:layout_height="match_parent"
   android:paddingLeft="40dp"
   android:orientation="vertical"
android:id="@+id/linearlayout" >
   <EditText
      android:id="@+id/txtName"
      android:layout_width="wrap_content"
      android:layout_height="wrap_content"
      android:layout_marginTop="25dp"
      android:ems="10"
      android:hint="Name"
      android:inputType="text"
      android:selectAllOnFocus="true" />
   EditText
      android:id="@+id/txtPwd"
      android:layout_width="wrap_content"
      android:layout_height="wrap_content"
      android:ems="10"
      android:hint="Password 0 to 9"

   android:inputType="numberPassword" />
    <EditText
      android:id="@+id/txtEmai"
      android:layout_width="wrap_content"
      android:layout_height="wrap_content"
      android:ems="10"
      android:hint="Email"

   android:inputType="textEmailAddress" />
```

Android – UI Control

Android EditText Example

```
<EditText
    android:id="@+id/txtDate"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_below="@+id/editText3"
    android:ems="10"
    android:hint="Date"
    android:inputType="date" />
  <EditText
    android:id="@+id/txtPhone"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:ems="10"
    android:hint="Phone Number"
    android:inputType="phone"
    android:textColorHint="#FE8DAB"/>
```

```
<<Button
    android:id="@+id/btnSend"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="submit"
    android:textSize="16sp"
    android:textStyle="normal|bold" />
  <TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:id="@+id/resultView"
    android:layout_marginTop="25dp"
    android:textSize="15dp"/>
</LinearLayout>
```


Android – UI Control

Android EditText Example

MainActivity.java

```
package com.tutlane.edittextexample;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.TextView;
import org.w3c.dom.Text;

public class MainActivity extends AppCompatActivity {
   Button btnSubmit;
   EditText name, password, email, dob, phoneno;
   TextView result;
   @Override
   protected void onCreate(Bundle savedInstanceState) {
      super.onCreate(savedInstanceState);
      setContentView(R.layout.activity_main);
      name=(EditText)findViewById(R.id.txtName);
      password = (EditText)findViewById(R.id.txtPwd);
      email = (EditText)findViewById(R.id.txtEmai);
```

Android – UI Control

Android EditText Example

```
 dob = (EditText)findViewById(R.id.txtDate);
phoneno= (EditText)findViewById(R.id.txtPhone);
btnSubmit = (Button)findViewById(R.id.btnSend);
result = (TextView)findViewById(R.id.resultView);

 btnSubmit.setOnClickListener(new View.OnClickListener() {
   @Override
   public void onClick(View v) {
      if (name.getText().toString().isEmpty() ||
          password.getText().toString().isEmpty() ||
          email.getText().toString().isEmpty() || dob.getText().toString().isEmpty()
             || phoneno.getText().toString().isEmpty()) {
         result.setText("Please Fill All the Details");
      } else {
         result.setText("Name -  " + name.getText().toString() + " \n" + "Password -  "
        + password.getText().toString() + " \n" + "E-Mail -  " +
        email.getText().toString() + " \n" + "DOB -  " + dob.getText().toString()
        + " \n" + "Contact -  " + phoneno.getText().toString());
```

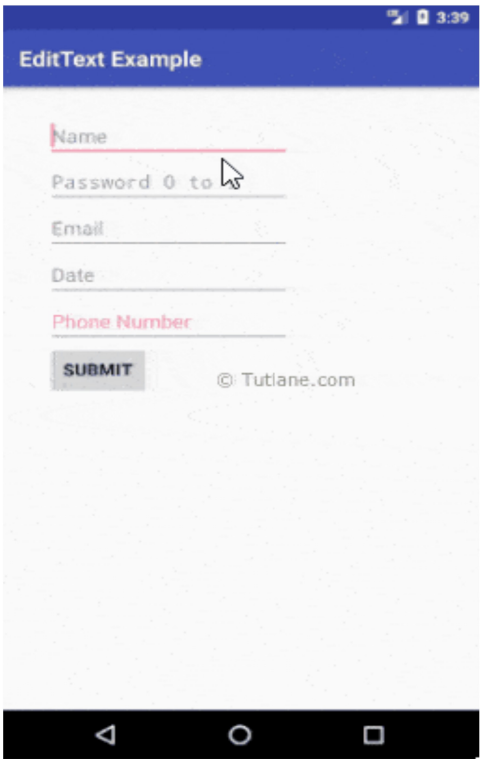

Android – UI Control

Android EditText Example

```
                }
        }
     });
  }
}
I
```

Android – UI Control

Output of Android EditText Example



Android – UI Control

Output of Android EditText Example (Once we enter details in all fields and click on Button we will get a result like as shown below.**)**


THANKS