Day 12: Controls: Input Controls, Part 1

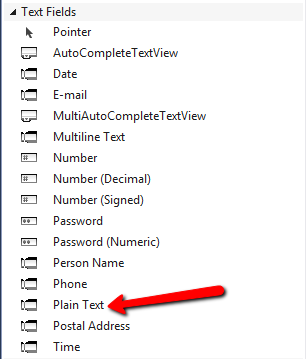
Over the course of next 5 days we will explore controls, or more commonly called in Android world as Views. In today’s post, let’s dig specifically into Input Controls, which are controls that lets users enter data or change state. We will explore the following controls –

1. EditText
   1. Date
   2. Email
   3. Multiline Text
   4. Number
   5. Password
   6. Person Name
   7. Phone
   8. Postal
2. CheckBox
3. RadioButton
4. ToggleButton
5. RatingBar

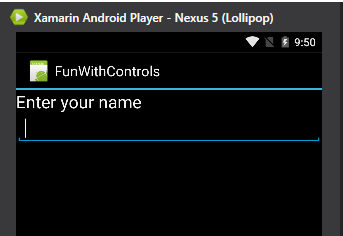
We will start today by exploring different type of EditText (Text Box) controls.

## EditText

EditText is the one of the most common input controls to enter text in Android. Open the Layout file that you are working with and then open Toolbox in Visual Studio to look for “Text Fields” section and “Plain Text” control.



Let’s drag and drop the Plain Text control on to the layout and run the application, it should have a simple Text Box like below –



Note: I added a TextView (Label) above, so the text field is a bit descriptive.

Now, let’s take a look at what AXML markup was generated as a result of dragging and dropping the EditText –

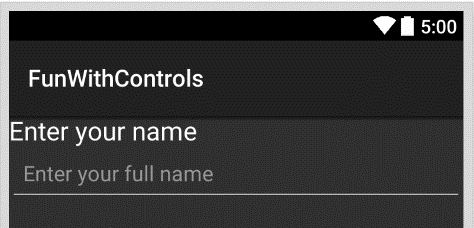
|  |
| --- |
| <?xml version="1.0" encoding="utf-8"?>  <LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"  android:orientation="vertical"  android:layout\_width="fill\_parent"  android:layout\_height="fill\_parent">  <TextView  android:text="Enter your name"  android:textAppearance="?android:attr/textAppearanceLarge"  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content"  android:id="@+id/textView1" />  <EditText  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content"  android:id="@+id/editText1" />  </LinearLayout> |

Gist file link: <https://gist.github.com/vkoppaka/55e993e7c8f6a61eb45d>

### Properties

You will notice the common android: attributes that we explored so far in the series like layout\_width, layout\_height, and id and nothing new. But, there is a huge repository of properties (attributes) you can add to the EditText, let’s explore a few –.

* android:hint – Adds place holder text to the EditText.
* android:padding – Adds padding inside the EditText through all four edges. You can still set individual Padding using the paddingBottom, paddingEnd, paddingLeft, paddingRight attributes.
* android:maxLength – Sets the maximum acceptable length of EditText. Once the length is hit, you cannot type any more characters into the EditText.
* android:inputType – When inputType is set to textAutoCorrect, Android suggests auto correct information to your users. There are more options here as well.
* android:enabled – Enables or Disables the EditText
* android:visibility – Shows or hides the EditText



Here is AXML for setting all the above EditText properties –

|  |
| --- |
| <?xml version="1.0" encoding="utf-8"?>  <LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"  android:orientation="vertical"  android:layout\_width="fill\_parent"  android:layout\_height="fill\_parent">  <TextView  android:text="Enter your name"  android:textAppearance="?android:attr/textAppearanceLarge"  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content"  android:id="@+id/textView1" />  <EditText  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content"  android:id="@+id/editText1"  android:hint="Enter your full name"  android:padding="12dp"  android:maxLength="5"  android:inputType="textAutoCorrect"  android:enabled="true"  android:visibility="visible" />  </LinearLayout> |

Gist file link: <https://gist.github.com/vkoppaka/a672f149b6743fa03460>

### Events

Now that we have explored properties of EditText, let’s take a look at few events of EditText –

* KeyPress – The event that gets fired as soon as user types something into the EditText.
* BeforeTextChanged, AfterTextChanged – Events that get fired before and after Text in EditText has been changed.
* Click – Event that gets fired as soon as you click on EditText.
* LongClick – Event that gets fired when the user presses and holds the EditText,
* TextChanged – Event that gets fired when the Text in the EditText gets changed.

|  |
| --- |
| using Android.App;  using Android.OS;  using Android.Views;  using Android.Widget;  namespace FunWithControls  {  [Activity(Label = "FunWithControls", MainLauncher = true, Icon = "@drawable/icon")]  public class MainActivity : Activity  {  protected override void OnCreate(Bundle bundle)  {  base.OnCreate(bundle);  // Set our view from the "main" layout resource  SetContentView(Resource.Layout.Main);  var editText = this.FindViewById<EditText>(Resource.Id.editText1);  editText.KeyPress += editText\_KeyPress;  }  void editText\_KeyPress(object sender, View.KeyEventArgs e)  {  e.Handled = true;  Toast.MakeText(this, "You typed something", ToastLength.Short);  }  }  } |

Gist file link: <https://gist.github.com/vkoppaka/9758d5271a12a205f861>

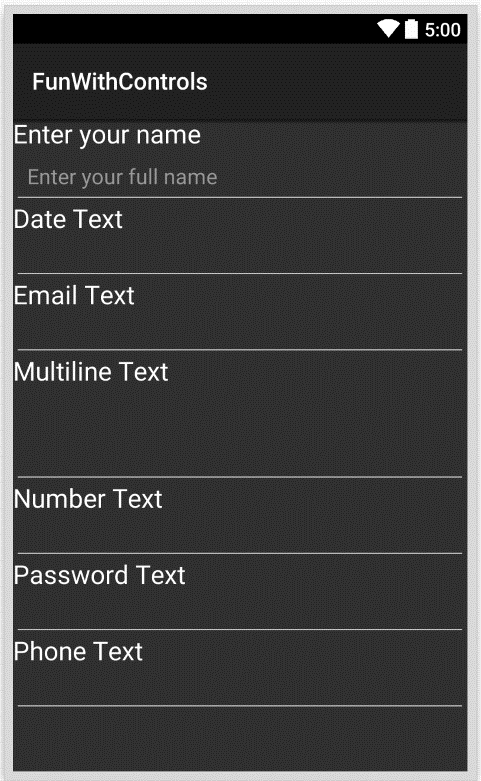
## Other kind of EditText

There are number of EditText types that come with Android. Every EditText differs mainly in inputType attribute. Let’s see what they are –

* Date Text – This text is used to enter Date data. android:inputType attribute is set to “date”. When the EditText is in this mode, user cannot enter any characters.
* Email Text – This text is used to gather Email data. android:inputType attribute is set to "textEmailAddress".
* Multineline Text – This text is used to gather more than one row of data from the user. android:minLines attribute is set to the number of lines you want the EditText to be.
* Number Text – This text is used to gather Number (digits) data from the user. android:inputType is set to "number" and the user cannot enter any characters into this EditText.
* Password Text – This text is used for Password fields. android:inputType is set to "textPassword" to mask the characters as they are being typed.

|  |
| --- |
| <?xml version="1.0" encoding="utf-8"?>  <LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"  android:orientation="vertical"  android:layout\_width="fill\_parent"  android:layout\_height="fill\_parent">  <TextView  android:text="Enter your name"  android:textAppearance="?android:attr/textAppearanceLarge"  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content"  android:id="@+id/textView1" />  <EditText  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content"  android:id="@+id/editText1"  android:hint="Enter your full name"  android:padding="12dp"  android:maxLength="5" />  <TextView  android:text="Date Text"  android:textAppearance="?android:attr/textAppearanceLarge"  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content"  android:id="@+id/textView2" />  <EditText  android:inputType="date"  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content"  android:id="@+id/editText2" />  <TextView  android:text="Email Text"  android:textAppearance="?android:attr/textAppearanceLarge"  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content"  android:id="@+id/textView3" />  <EditText  android:inputType="textEmailAddress"  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content"  android:id="@+id/editText3" />  <TextView  android:text="Multiline Text"  android:textAppearance="?android:attr/textAppearanceLarge"  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content"  android:id="@+id/textView4" />  <EditText  android:inputType="textMultiLine"  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content"  android:id="@+id/editText4"  android:minLines="3" />  <TextView  android:text="Number Text"  android:textAppearance="?android:attr/textAppearanceLarge"  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content"  android:id="@+id/textView5" />  <EditText  android:inputType="number"  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content"  android:id="@+id/editText5" />  <TextView  android:text="Password Text"  android:textAppearance="?android:attr/textAppearanceLarge"  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content"  android:id="@+id/textView6" />  <EditText  android:inputType="textPassword"  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content"  android:id="@+id/editText6" />  <TextView  android:text="Phone Text"  android:textAppearance="?android:attr/textAppearanceLarge"  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content"  android:id="@+id/textView7" />  <EditText  android:inputType="phone"  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content"  android:id="@+id/editText7" />  </LinearLayout> |

Gist file link: <https://gist.github.com/vkoppaka/837a566fe29420366451>



Now that we explored EditText in detail, tomorrow we will focus on few more Input Controls in Android.

Venkata