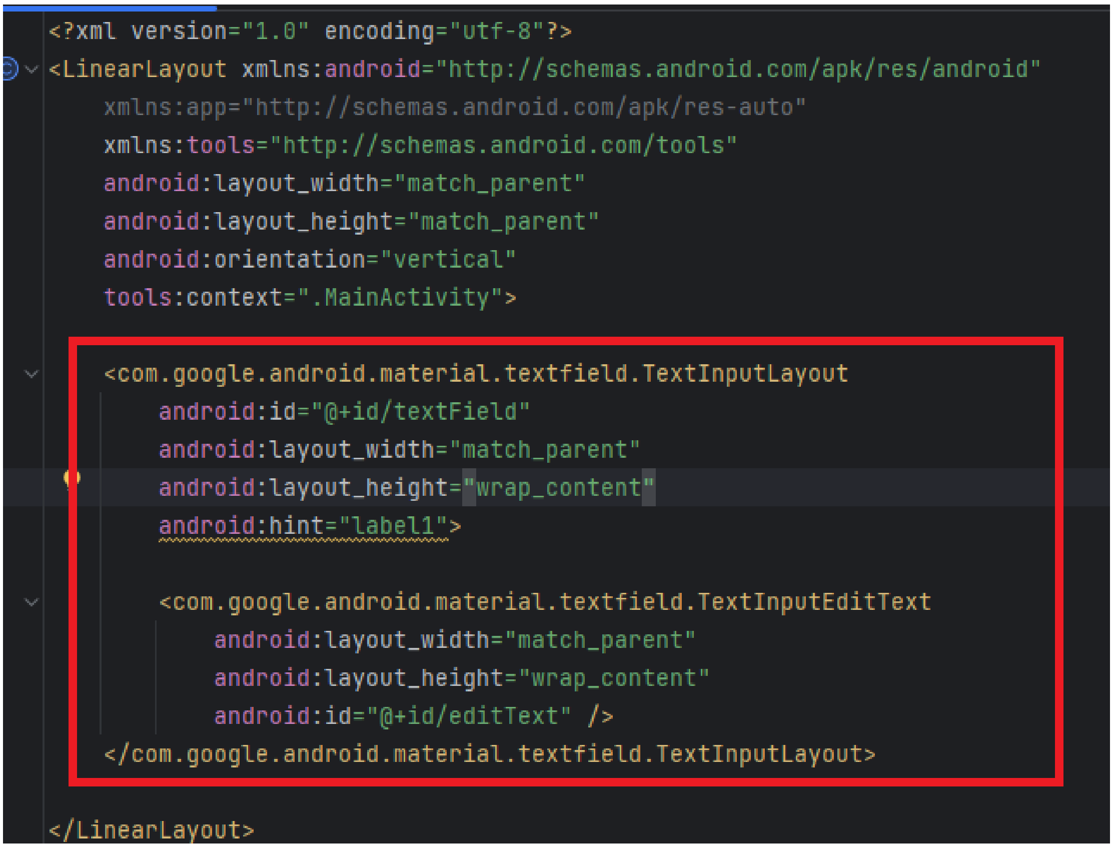
Source code address

material-components-android/docs/components/TextField.md at master · material-components/material-components-android (github.com)

Steps

TextField offers superior functionality and style compared to the traditional EditText and is easier to implement.

1. Simply copy the code into main_activity.xml



```xml
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout 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"
    android:orientation="vertical"
    tools:context=".MainActivity">

    <com.google.android.material.textfield.TextInputLayout
        android:id="@+id/textField"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:hint="label1">

        <com.google.android.material.textfield.TextInputEditText
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:id="@+id/editText" />
    </com.google.android.material.textfield.TextInputLayout>

</LinearLayout>
```

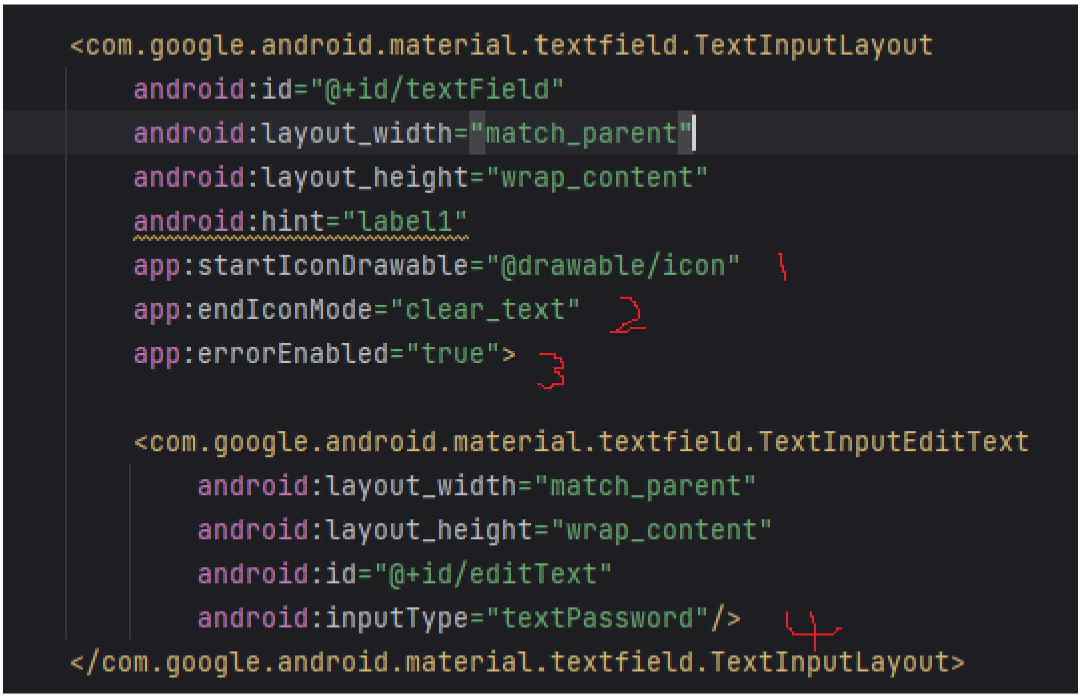
2. Several commonly used attributes include:

1) Leading icon.

2) Trailing icon (which can clear the text field).

3) Error prompt (text field changes color and displays error text).

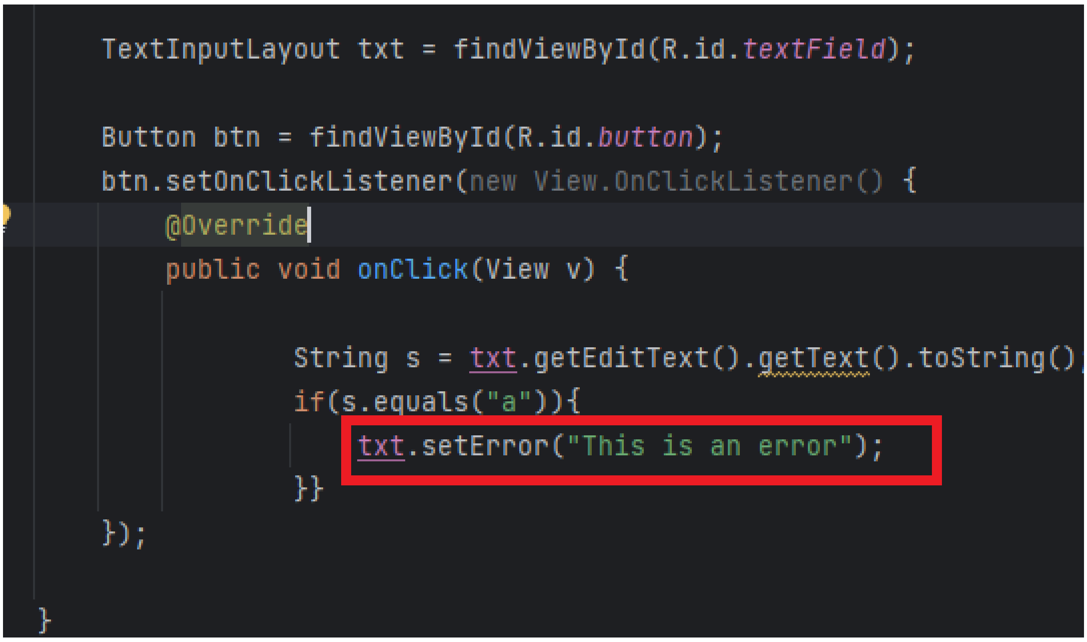
4) In TextInputEditText, the input type (such as password) can be set.



```xml
<com.google.android.material.textfield.TextInputLayout
    android:id="@+id/textField"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:hint="label1"
    app:startIconDrawable="@drawable/icon"
    app:endIconMode="clear_text"
    app:errorEnabled="true">

    <com.google.android.material.textfield.TextInputEditText
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:id="@+id/editText"
        android:inputType="textPassword"/>
</com.google.android.material.textfield.TextInputLayout>
```

3. In Java code, instantiate TextInputLayout directly; it has a getEditText() method that can locate the EditText. As shown, when we enter "a" and click the button, the TextFields will turn red to indicate an error.



```java
    TextInputLayout txt = findViewById(R.id.textField);

    Button btn = findViewById(R.id.button);
    btn.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {

                String s = txt.getEditText().getText().toString()
                if(s.equals("a")){
                    txt.setError("This is an error");
                }}
    });

}
```