

50% off yearly subscriptions, for a limited time! [FIND OUT MORE](#)



CODE > ANDROID SDK

Create a Custom Keyboard on Android

by Ashraff Hathibelagal 1 Dec 2014

Difficulty: Beginner Length: Medium Languages: English ▾

Android SDK Java Mobile Development Eclipse IDEs



What You'll Be Creating

Most Android devices don't have a physical keyboard. Instead, they rely on a virtual or soft keyboard to accept user input. If you're into Android personalization, knowing how to build a custom, soft keyboard can take your hobby to a whole new level.

Using the Android SDK, you can quickly create a soft keyboard with surprisingly

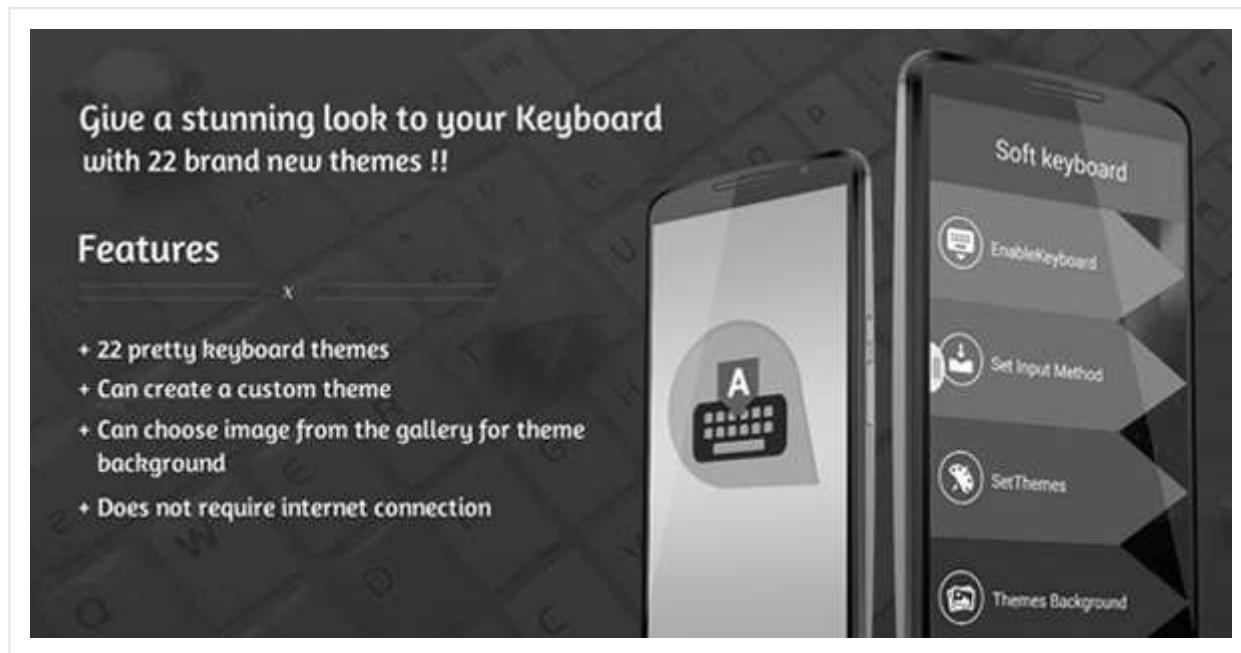
few lines of code, because the SDK takes care of a lot of the low level tasks, such as recognizing key touches, drawing the keyboard, and establishing connections between the keyboard and input fields.

In this tutorial, you will learn how to create a fully functional soft keyboard that can serve as your Android device's default keyboard.

Premium Option

If you're in a hurry, check out [Android Keyboard Themes](#), a ready-to-use solution from Envato Market.

The app gives you the flexibility to choose one of the 22 built-in keyboard themes or create your own custom theme.



Android Keyboard Themes

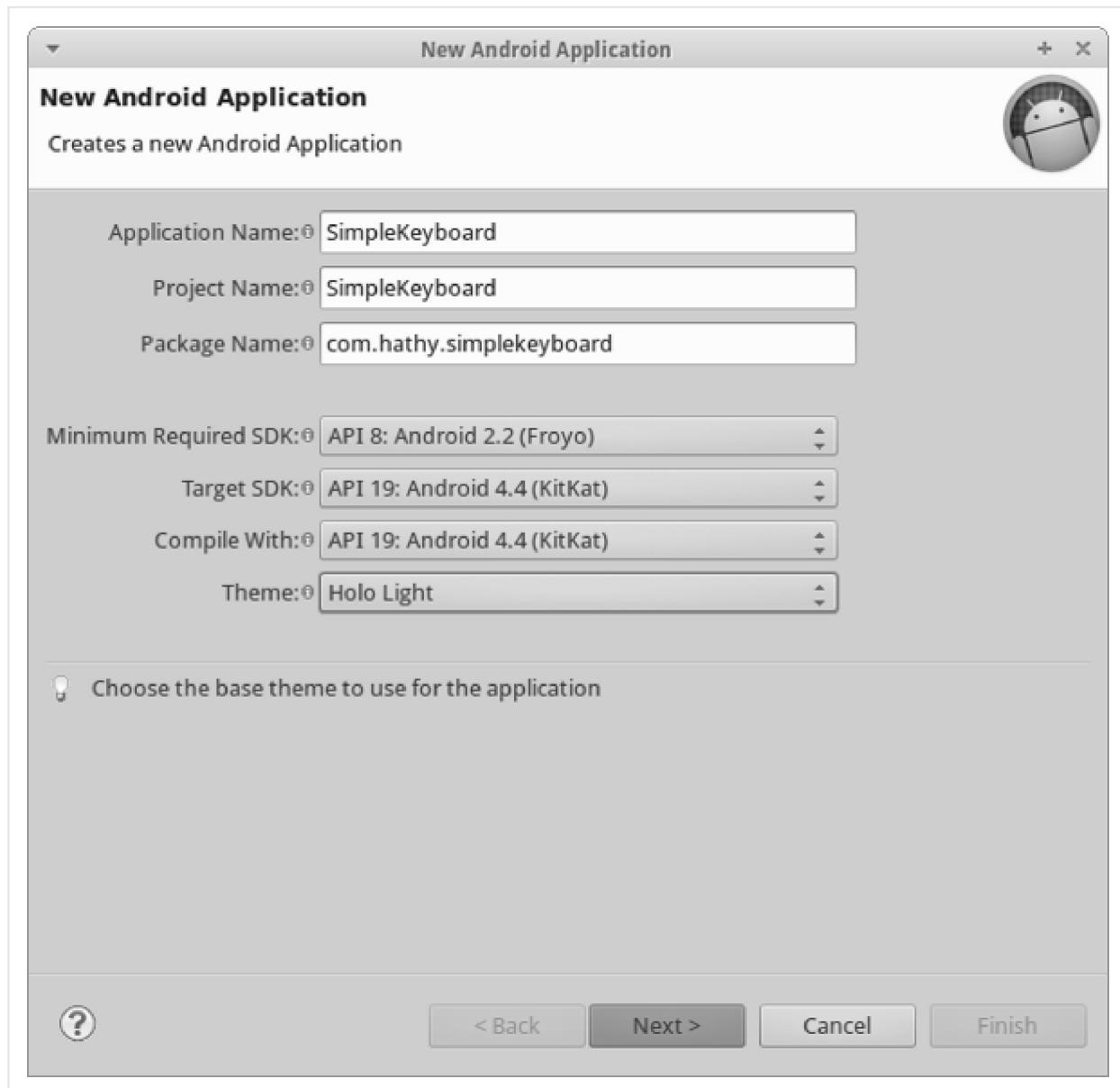
If you prefer to build your own, read on to find out how.

1. Prerequisites

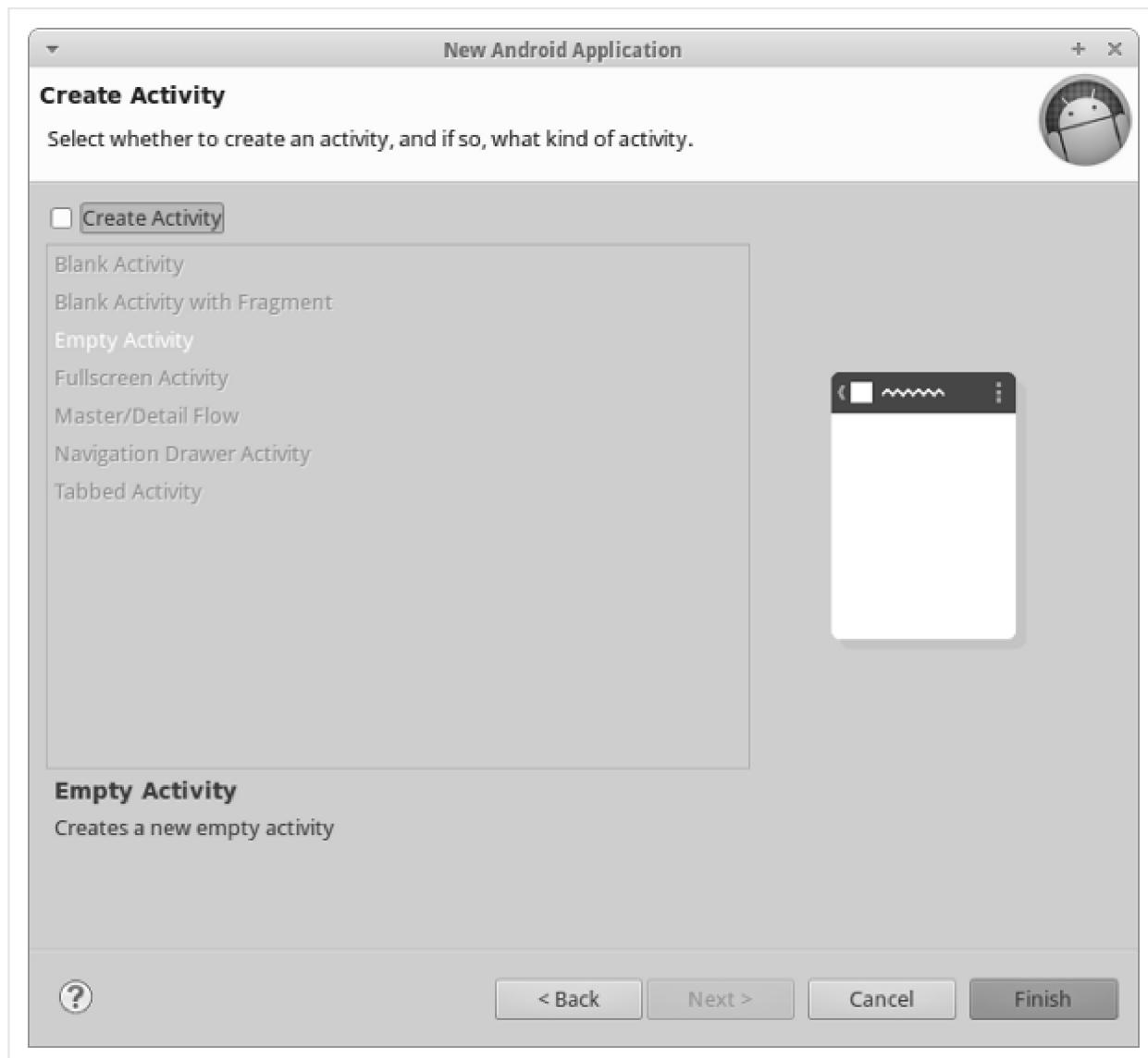
You will need the Eclipse ADT Bundle installed. You can download it from the [Android Developer website](#).

2. Create a New Project

Fire up Eclipse and create a new Android application. Call this application, **SimpleKeyboard**. Make sure you choose a unique package name. Set the minimum required SDK to **Android 2.2** and set the target SDK to **Android 4.4**.



This app will have no activities so deselect **Create Activity** and click **Finish**.



3. Edit the Manifest

A soft keyboard is considered as an **Input Method Editor (IME)** by the Android operating system. An IME is declared as a `Service` in `AndroidManifest.xml` that uses the `BIND_INPUT_METHOD` permission, and responds to the action `android.view.InputMethod`.

Add the following lines to the `application` tag of the manifest:

```
1 <service android:name=".SimpleIME"
2     android:label="@string/simple_ime"
3     android:permission="android.permission.BIND_INPUT_METHOD"
4     >
5     <meta-data android:name="android.view.im" android:resource="@xml/method"/>
6     <intent-filter>
7         <action android:name="android.view.InputMethod" />
8     </intent-filter>
9 </service>
```

4. Create method.xml

The `service` tag in the manifest file contains a `meta-data` tag that references an XML file named **method.xml**. Without this file, the Android operating system won't recognize our `Service` as a valid IME service. The file contains details about the input method and its subtypes. For our keyboard, we define a single subtype for the **en_US** locale. Create the directory **res/xml** if it doesn't exist, and add the file **method.xml** to it. The contents of the file should be:

```

1 <?xml version="1.0" encoding="utf-8"?>
2 <input-method xmlns:android="http://schemas.android.com/apk/res/android">
3     <subtype
4         android:label="@string/subtype_en_US"
5         android:imeSubtypeLocale="en_US"
6         android:imeSubtypeMode="keyboard" />
7 </input-method>
```

5. Edit strings.xml

The strings that this app uses are defined in the **res/values/strings.xml** file. We're going to need three strings:

- the name of the app
- the label of the IME
- the label of the IME's subtype

Update your **strings.xml** so that it has the following contents:

```

1 <resources>
2     <string name="app_name">SimpleKeyboard</string>
3     <string name="simple_ime">Simple IME</string>
4     <string name="subtype_en_US">English (US)</string>
5 </resources>
```

6. Define the Keyboard Layout

The layout of our keyboard contains only a `KeyboardView`. The `layout_alignParentBottom` attribute is set to `true` so that keyboard appears at the bottom of the screen.

Create a file named **res/layout/keyboard.xml** and replace its contents with the following:

```
1 <?xml version="1.0" encoding="UTF-8"?>
2 <android.inputmethodservice.KeyboardView
3     xmlns:android="http://schemas.android.com/apk/res/android"
4     android:id="@+id/keyboard"
5     android:layout_width="match_parent"
6     android:layout_height="wrap_content"
7     android:layout_alignParentBottom="true"
8     android:keyPreviewLayout = "@layout/preview"
9 />
```

The `keyPreviewLayout` is the layout of the short-lived pop-up that shows up whenever a key on the keyboard is pressed. It contains a single `TextView`. Create a file named **res/layout/preview.xml** and add the following to it:

```
01 <?xml version="1.0" encoding="utf-8"?>
02 <TextView xmlns:android="http://schemas.android.com/apk/res/android"
03     android:layout_width="match_parent"
04     android:layout_height="match_parent"
05     android:gravity="center"
06     android:background="#fffff00"
07     android:textStyle="bold"
08     android:textSize="30sp"
09     >
10 </TextView>
```

6. Define the Keyboard Keys

The details of the keyboard keys and their positions are specified in an XML file. Every key has the following attributes:

- `keyLabel` : This attribute contains the text that is displayed on the key.
- `codes` : This attribute contains the unicode values of the characters that the key represents.

For example, to define a key for the letter **A**, the `codes` attribute should have the value **97** and the `keyLabel` attribute should be set to **A**.

If more than one code is associated with a key, then the character that the key represents will depend on the number of taps the key receives. For example, if a key

has the codes **63**, **33**, and **58**:

- a single tap on the key results in the character ?
- two taps in quick succession results in the character !
- three taps in quick succession results in the character :

A key can also have a few optional attributes:

- `keyEdgeFlags` : This attribute can take the value `left` or `right`. This attribute is usually added to the leftmost and rightmost keys of a row.
- `keyWidth` : This attribute defines the width of a key. It's usually defined as a percentage value.
- `isRepeatable` : If this attribute is set to `true`, long-pressing the key will repeat the action of the key multiple times. It is usually set to `true` for the delete and spacebar keys.

The keys of a keyboard are grouped as rows. It's good practice to limit the number of keys on a row to a maximum of ten, with each key having a width equal to 10% of the keyboard. The height of the keys is set to **60dp** in this tutorial. This value can be adjusted, but values less than **48dp** are not recommended. Our keyboard will have five rows of keys.

We can now go ahead and design the keyboard. Create a new file named **res/xml/qwerty.xml** and replace its contents with the following:

```

01 <Keyboard xmlns:android="http://schemas.android.com/apk/res/android"
02     android:keyWidth="10%p"
03     android:horizontalGap="0px"
04     android:verticalGap="0px"
05     android:keyHeight="60dp"
06 >
07     <Row>
08         <Key android:codes="49" android:keyLabel="1" android:keyEdgeFlags="left"/>
09         <Key android:codes="50" android:keyLabel="2"/>
10         <Key android:codes="51" android:keyLabel="3"/>
11         <Key android:codes="52" android:keyLabel="4"/>
12         <Key android:codes="53" android:keyLabel="5"/>
13         <Key android:codes="54" android:keyLabel="6"/>
14         <Key android:codes="55" android:keyLabel="7"/>
15         <Key android:codes="56" android:keyLabel="8"/>
16         <Key android:codes="57" android:keyLabel="9"/>
17         <Key android:codes="48" android:keyLabel="0" android:keyEdgeFlags="right"/>
18     </Row>
19     <Row>
20         <Key android:codes="113" android:keyLabel="q" android:keyEdgeFlags="left"/>
```

```

21 <Key android:codes="119" android:keyLabel="w"/>
22 <Key android:codes="101" android:keyLabel="e"/>
23 <Key android:codes="114" android:keyLabel="r"/>
24 <Key android:codes="116" android:keyLabel="t"/>
25 <Key android:codes="121" android:keyLabel="y"/>
26 <Key android:codes="117" android:keyLabel="u"/>
27 <Key android:codes="105" android:keyLabel="i"/>
28 <Key android:codes="111" android:keyLabel="o"/>
29 <Key android:codes="112" android:keyLabel="p" android:keyEdgeFlags="right"/>
30 </Row>
31 <Row>
32 <Key android:codes="97" android:keyLabel="a" android:keyEdgeFlags="left"/>
33 <Key android:codes="115" android:keyLabel="s"/>
34 <Key android:codes="100" android:keyLabel="d"/>
35 <Key android:codes="102" android:keyLabel="f"/>
36 <Key android:codes="103" android:keyLabel="g"/>
37 <Key android:codes="104" android:keyLabel="h"/>
38 <Key android:codes="106" android:keyLabel="j"/>
39 <Key android:codes="107" android:keyLabel="k"/>
40 <Key android:codes="108" android:keyLabel="l"/>
41 <Key android:codes="35,64" android:keyLabel="\# \@" android:keyEdgeFlags="ri
42 </Row>
43 <Row>
44 <Key android:codes="-1" android:keyLabel="CAPS" android:keyEdgeFlags="left"/>
45 <Key android:codes="122" android:keyLabel="z"/>
46 <Key android:codes="120" android:keyLabel="x"/>
47 <Key android:codes="99" android:keyLabel="c"/>
48 <Key android:codes="118" android:keyLabel="v"/>
49 <Key android:codes="98" android:keyLabel="b"/>
50 <Key android:codes="110" android:keyLabel="n"/>
51 <Key android:codes="109" android:keyLabel="m"/>
52 <Key android:codes="46" android:keyLabel="."/>
53 <Key android:codes="63,33,58" android:keyLabel="\? ! :" android:keyEdgeFlags
54 </Row>
55 <Row android:rowEdgeFlags="bottom">
56 <Key android:codes="44" android:keyLabel="," android:keyWidth="10%p" androi
57 <Key android:codes="47" android:keyLabel="/" android:keyWidth="10%p" />
58 <Key android:codes="32" android:keyLabel="SPACE" android:keyWidth="40%p" and
59 <Key android:codes="-5" android:keyLabel="DEL" android:keyWidth="20%p" andro
60 <Key android:codes="-4" android:keyLabel="DONE" android:keyWidth="20%p" andr
61 </Row>
62 </Keyboard>

```

You may have noticed that some keys have negative values for the `codes` attribute. Negative values are equal to predefined constants in the `Keyboard` class. For example, the value `-5` is equal to the value of `Keyboard.KEYCODE_DELETE`.

7. Create a `Service` Class

Create a new Java class and call it **SimpleIME.java**. The class should extend `InputMethodService` class and implement

the `OnKeyboardActionListener` interface. The `OnKeyboardActionListener` interface contains the methods that are called when keys of the soft keyboard are tapped or pressed.

The `SimpleIME` class should have three member variables:

- a `KeyboardView` referencing the view defined in the layout
- a `Keyboard` instance that is assigned to the `KeyboardView`
- a `boolean` telling us if the caps lock is enabled

After declaring these variables and adding the methods of the `OnKeyboardActionListener` interface, the `SimpleIME` class should look like this:

```
01 public class SimpleIME extends InputMethodService
02     implements OnKeyboardActionListener{
03
04     private KeyboardView kv;
05     private Keyboard keyboard;
06
07     private boolean caps = false;
08
09     @Override
10     public void onKey(int primaryCode, int[] keyCodes) {
11
12     }
13
14     @Override
15     public void onPress(int primaryCode) {
16     }
17
18     @Override
19     public void onRelease(int primaryCode) {
20     }
21
22     @Override
23     public void onText(CharSequence text) {
24     }
25
26     @Override
27     public void swipeDown() {
28     }
29
30     @Override
31     public void swipeLeft() {
32     }
33
34     @Override
35     public void swipeRight() {
36     }
37
38     @Override
39     public void swipeUp() {
40     }
41 }
```

When the keyboard is created, the `onCreateInputView` method is called. All the member variables of the `Service` can be initialized here. Update the implementation of the `onCreateInputView` method as shown below:

```

1  @Override
2  public View onCreateInputView() {
3      kv = (KeyboardView)getLayoutInflater().inflate(R.layout.keyboard, null);
4      keyboard = new Keyboard(this, R.xml.qwerty);
5      kv.setKeyboard(keyboard);
6      kv.setOnKeyboardActionListener(this);
7      return kv;
8  }

```

Next, we create a method that plays a sound when a key is pressed. We use the `AudioManager` class to play the sounds. The Android SDK includes a few default sound effects for key presses and those are used in the `playClick` method.

```

01  private void playClick(int keyCode){
02      AudioManager am = (AudioManager) getSystemService(AUDIO_SERVICE);
03      switch(keyCode){
04          case 32:
05              am.playSoundEffect(AudioManager.FX_KEYPRESS_SPACEBAR);
06              break;
07          case Keyboard.KEYCODE_DONE:
08          case 10:
09              am.playSoundEffect(AudioManager.FX_KEYPRESS_RETURN);
10              break;
11          case Keyboard.KEYCODE_DELETE:
12              am.playSoundEffect(AudioManager.FX_KEYPRESS_DELETE);
13              break;
14          default: am.playSoundEffect(AudioManager.FX_KEYPRESS_STANDARD);
15      }
16  }

```

Finally, update the `onKey` method so that our keyboard app can communicate with input fields (usually `EditText` views) of other applications.

The `getCurrentInputConnection` method is used to get a connection to the input field of another application. Once we have the connection, we can use the following methods:

- `commitText` to add one or more characters to the input field
- `deleteSurroundingText` to delete one or more characters of the input field
- `sendKeyEvent` to send events, like `KEYCODE_ENTER`, to the external application

Whenever a user presses a key on the soft keyboard, the `onKey` method is called with

the unicode value of the key as one of its parameters. Based on this value, the keyboard performs one of the following actions:

- If the code is `KEYCODE_DELETE`, one character to the left of the cursor is deleted using the `deleteSurroundingText` method.
- If the code is `KEYCODE_DONE`, a `KEYCODE_ENTER` key event is fired.
- If the code is `KEYCODE_SHIFT`, the value of the `caps` variable is changed and the shift state of the keyboard is updated using the `setShifted` method. The keyboard needs to be redrawn when the state changes so that the labels of the keys are updated. The `invalidateAllKeys` method is used to redraw all keys.
- For all other codes, the code is simply converted into a character and sent to the input field. If the code represents a letter of the alphabet and the `caps` variable is set to `true`, then the character is converted to uppercase.

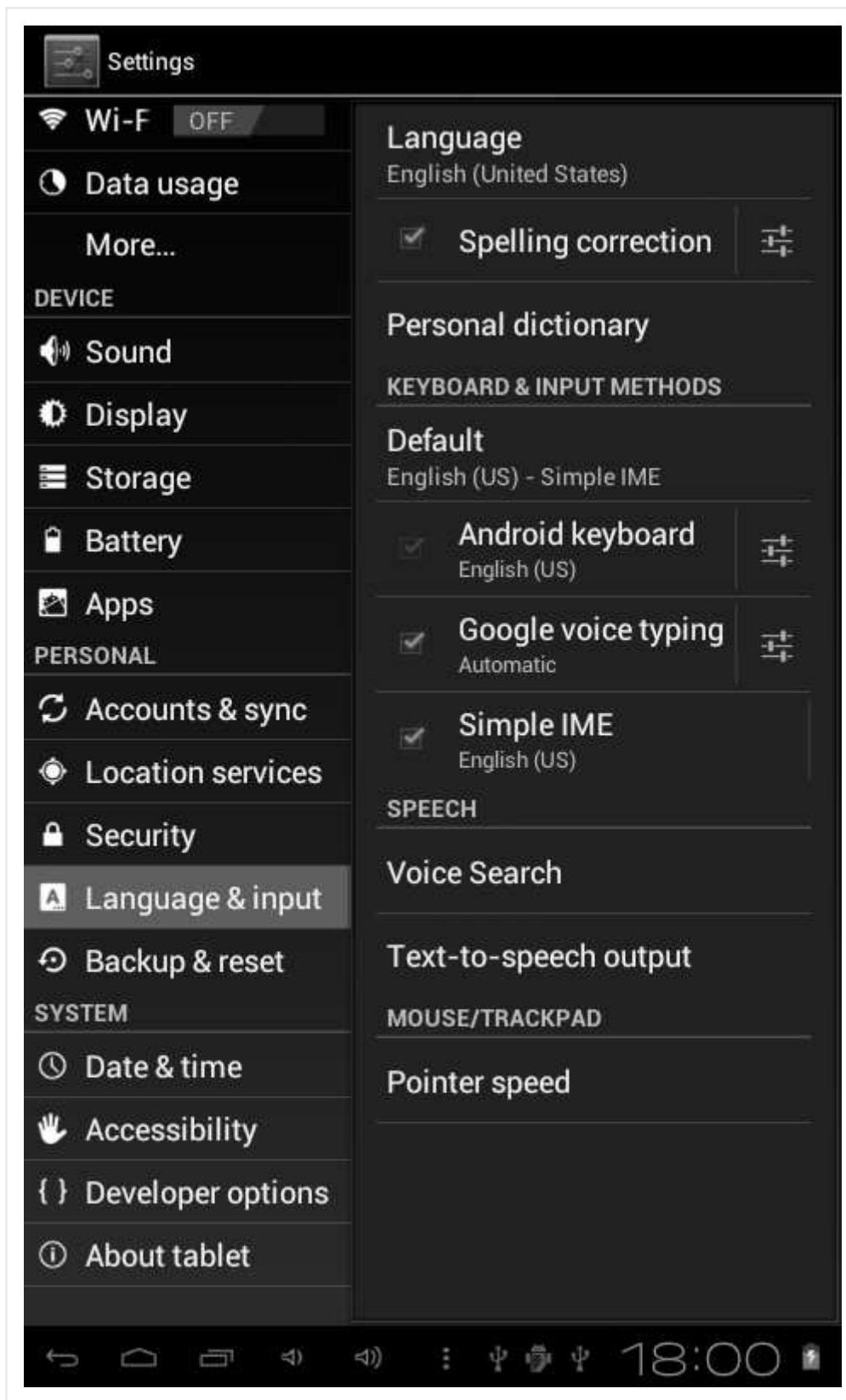
Update the `onKey` method so that it looks like this:

```
01  @Override
02  public void onKey(int primaryCode, int[] keyCodes) {
03      InputConnection ic = getCurrentInputConnection();
04      playClick(primaryCode);
05      switch(primaryCode){
06          case Keyboard.KEYCODE_DELETE :
07              ic.deleteSurroundingText(1, 0);
08              break;
09          case Keyboard.KEYCODE_SHIFT:
10              caps = !caps;
11              keyboard.setShifted(caps);
12              kv.invalidateAllKeys();
13              break;
14          case Keyboard.KEYCODE_DONE:
15              ic.sendKeyEvent(new KeyEvent(KeyEvent.ACTION_DOWN, KeyEvent.KEYCODE_ENTER));
16              break;
17          default:
18              char code = (char)primaryCode;
19              if(Character.isLetter(code) && caps){
20                  code = Character.toUpperCase(code);
21              }
22              ic.commitText(String.valueOf(code),1);
23      }
24 }
```

8. Testing the Keyboard

The soft keyboard is now ready to be tested. Compile and run it on an Android device.

This app doesn't have an **Activity**, which means that it won't show up in the launcher. To use it, it should first be activated in the device's **Settings**.



After activating **Simple IME**, open any app that allows text input (for example, any messaging app) and click on one of its input fields. You should see a keyboard icon appear in the notifications area. Depending on your device, you can either click on that

icon or drag the notification bar down and select **Simple IME** as the input method. You should now be able to type using your new keyboard.



Conclusion

In this tutorial, you have learned how to create a custom keyboard app from scratch. To change the look and feel of your keyboard, all you have to do is add extra styling to the **res/layout/keyboard.xml** and **res/layout/preview.xml** files. To change the positions of

the keys, update the **res/xml/qwerty.xml** file. To add more features to your keyboard, refer to the developer documentation.



Ashraff Hathibelagal

Hathibelagal is an independent Android app developer and blogger who loves tinkering with new frameworks, SDKs and devices.



tuts+

**STUDENT ACCESS
JUST \$90/YR**

Courses, eBooks & more >

[View on Github](#)

Translations

Envato Tuts+ tutorials are translated into other languages by our community members—you can be involved too!

[Translate this post](#)

Powered by  native

[112 Comments](#)[Tuts+ Hub](#)[Login](#)[Heart](#) Recommend 21[Share](#)[Sort by Best](#)[Join the discussion...](#)[Kirshan Luhana](#) • 2 years ago

I am sorry, but I can not run this code, can I download code from any source ?

[8](#) [^](#) | [v](#) • [Reply](#) • [Share](#)[Igor Ganapolsky](#) ➔ Kirshan Luhana • 7 months ago

You can check out the source here: http://androidxref.com/6.0.0_r...

[^](#) | [v](#) • [Reply](#) • [Share](#)



Meher Baloch → Igor Ganapolsky • 6 months ago

going to make keyboard for my native language....suffering from same problem , wanna contact u whats app or any other contact ...

^ | v • Reply • Share >



Kirshan Luhana → Igor Ganapolsky • 7 months ago

Thank you very much, I will check it soon

^ | v • Reply • Share >



Avaholic • 2 years ago

Good tutorial! Now, I've been wondering something. I'm creating a custom keyboard like this, but I want to be able to output emoticons (or other images) to the selected EditText. I know it's possible, because Emoji does it with their keyboard. They've somehow got the ability to add emoticons to any EditText. In all my searches across the internet, I cannot find anyone who knows how they did it. Do you know?

And just so you know, creating a SpannableStringBuilder, then setting an ImageSpan on it, then committing that using InputConnection doesn't work. I've heard that people often instead of just create a custom view instead of using an actual custom keyboard by extending InputMethodService (as you've done here), but how would I actually use that as a Keyboard?

10 ^ | v • Reply • Share >



Lucas Zorzi • a year ago

PROBLEM? To solve the run problem, change the MANIFEST.XML by this:

```
<manifest xmlns:android="http://schemas.android.com/apk..." package="com.simplekeyboard">

<application android:label="@string/simple_ime">
<service android:name="SimpleIME"
        android:permission="android.permission.BIND_INPUT_METHOD">

<intent-filter>
<action android:name="android.view.InputMethod"/>
</intent-filter>
<meta-data android:name="android.view.im" android:resource="@xml/method"/>
</service>
</application>
</manifest>
```

!!! The url on first line is ...apk/res/android"

!!! Change to your package on the 2º line

Now the keyboard appears in the menu "Language and input" on the configurations!

DALE BR!!

2 ^ | v • Reply • Share >



Kirshan Luhana • 2 years ago

can you explain in details on 7th section (Creating service) I am getting errors

package SimpleIME.SimpleIME;

```

import android.app.Service;
import android.content.Intent;
import android.os.IBinder;

public class SimpleIME extends InputMethodService
    implements OnKeyboardActionListener{

    private KeyboardView kv;
    private Keyboard keyboard;

    private boolean caps = false;

    @Override
    public void onPress(int primaryCode) {
    }

```

[see more](#)[3 ^ | v • Reply • Share >](#)**el Demoledor** → Kirshan Luhana • 2 years ago

me too

[1 ^ | v • Reply • Share >](#)**PrakashK Gowin** → el Demoledor • a year ago

I did it, without any error. it is good

[^ | v • Reply • Share >](#)**Rajan** → PrakashK Gowin • 4 months ago

How can you do that i have a problem in manifest file .plese help me . i mention my problem here .

[^ | v • Reply • Share >](#)**Guest** • 2 years agothis one much better : <https://android.googlesource.c...>[3 ^ | v • Reply • Share >](#)**Luis de la Orden** → Guest • 2 years ago

Man, there is a way of suggesting alternatives that actually makes the person suggesting look good, professional and not a complete a-hole even if the link on the other side turns out to be completely useless, which is the case with your link, Mark. You don't seem to grasp the basic difference between a bunch of code samples and a tutorial.

As a tutorial, this is the best I have seen so far, the author has done a great job by taking his time to write everything down and setting it up in steps that are easy to follow, saving many of us from having to shell out thousands of dollars to create keyboards for languages that have no representation in the digital world yet.

Please expand your tutorial by teaching how to add the pop up letters with accents and keep the great job, you have talent and as a result of your work I will hopefully create a few

native keyboards. I will keep you posted!

Native keyboards, I will keep you posted!

2 ^ | v • Reply • Share >



anon → Luis de la Orden • 2 years ago

I want to change the keys to different letters in my local language when user clicks on shift.

please share how to do it?

^ | v • Reply • Share >



Luis de la Orden → anon • 2 years ago

Anon, I got stuck half-way through. I believe we all need to work together as individually we are just scattering efforts.

I created a repository in GitHub hoping we can get more people working together to produce a complete template framework for any local language.

[https://github.com/luismorais/...](https://github.com/luismorais/)

^ | v • Reply • Share >



stupido → Luis de la Orden • 2 years ago

Instead of coping and pasting everything try re-typing the items that are causing errors - it needs you to input them by hand so it can find and import them.

I'm pretty close to getting it to work but not just yet :)

2 ^ | v • Reply • Share >



Meher Baloch → stupido • 6 months ago

going to make keyboard for my native language....suffering from same problem , wanna contact u whats app or any other contact ...

^ | v • Reply • Share >



Luis de la Orden → stupido • 2 years ago

Hi stupido!

Really happy to hear you are close to putting the whole thing together. I would be very interested to see what you achieve and get your name into the localisation initiative!

^ | v • Reply • Share >



Meher Baloch → Luis de la Orden • 6 months ago

going to make keyboard for my native language....suffering from same problem , whats app or any other contact...

^ | v • Reply • Share >



Cillendor • 10 months ago

I want to create a custom keyboard that supports a Graphite font, Tengwar Telcontar. Is this possible? I don't think Android has Graphite capabilities yet... But in general, how does one use custom fonts for this so as to build in another alphabet?

1 ^ | v • Reply • Share >



Brittany • a month ago

I get no errors when I build and run the app but it does not show up on my phone or emulator. Any suggestions?

^ | v • Reply • Share >



Rajveer Singh • a month ago

```
kv = (KeyboardView)getLayoutInflater().inflate(R.layout.keyboard, null);
keyboard = new Keyboard(this, R.xml.qwerty);
unable to get Resources
```

^ | v • Reply • Share >



izumi • a month ago

hi I have a trouble about how to show different height key (like the pic). I have tried to modify key height, but the result is not ok, like the second picture... who knows how to fix the problem? Please help...thanks!

I write the keyboard layout as:

```
<keyboard android:horizontalgap="1dp" android:keyheight="54dp" android:keywidth="20%p"
    android:verticalgap="1dp">

<row>

    <key android:codes="49" android:keyedgeflags="left" android:keylabel="1"/>

    <key android:codes="50" android:keylabel="2"/>

    <key android:background="@color/c5" android:codes="51" android:keylabel="3"/>

    <key android:codes="-4" android:isrepeatable="true" android:keyedgeflags="right"
        android:keyheight="108dp" android:keyicon="@drawable/keyboard_delete_bg"
        android:keywidth="40%p"/>
```

[see more](#)

^ | v • Reply • Share >



Vijayadhas • a month ago

How to create virtual keyboard like External Keyboard helper?? Please let me any idea...

^ | v • Reply • Share >



Brad Smeltzer • 3 months ago

Dear friends,

I am having trouble. Using Android Studio 2.1 I get the following error:

Could not identify launch activity: Default Activity not found
Error while Launching activity

Also, in the AndroidManifest.xml there seems to be an error:
service android:name=".SimpleIMF"

SimpleIMF is in red indicating something is wrong I believe

Comment is in red, indicating something is wrong, I believe.

The app installs, but crashes when chosen.

I went over and over the procedure but can't find an error. Can anyone help? Thanks!

^ | v • Reply • Share >



Brittany → Brad Smeltzer • 20 days ago

I just figured out what needed to be done.

You have to edit the configurations and change the launch option to "nothing". This way it won't try to launch an activity that does not exist.

run->edit configurations then look for the launch options

^ | v • Reply • Share >



Hasan Albanna • 3 months ago

Do you, by chance, still maintain this post? It is a very useful tutorial. It helps me a lot in handling with my cracked tab (some part of the screen is no longer responsive) so I was able to adjust the position of the keyboard by editing the xml. Just one issue though: the done button (Keyboard.KEYCODE_DONE) does not work in browser (i.e. after entering the address, pressing 'done' does not direct me to the site). It works in all other textviews but in browser.

^ | v • Reply • Share >



Sudipta Ranjan Dash • 4 months ago

Sir i want to add my own prediction algorithm in this keyboard.so where should i add ? in the softkeyboard above of keyboard a white space is coming ,where it resides in the source code?

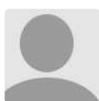
^ | v • Reply • Share >



Prakaash M • 4 months ago

I have created this project successfully as said. I tried to install on my phone apk was installed, but it doesn't show up in my Input methods in my phone settings. What should I do to make it visible in my phone's Input methods.

^ | v • Reply • Share >



Naman Bhatia • 4 months ago

How can I open a new keyboard from the existing one for the special characters?

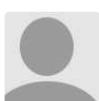
^ | v • Reply • Share >



Rajan • 4 months ago

how can I set image view at the place of keyboard.send me source code.or help me .need it .

^ | v • Reply • Share >



Rajan • 4 months ago

this is my manifest file .how can I use the service into the activity.

```
<application android:allowBackup="true" android:icon="@mipmap/ic_launcher"  
    android:label="@string/app_name" android:supportsRtl="true"  
    android:theme="@style/AppTheme">
```

```
<activity android:name=".SimpleIME">
```

```
<intent-filter>  
  
<action android:name="android.intent.action.MAIN"/>  
  
<category android:name="android.intent.category.LAUNCHER"/>  
  
</intent-filter>  
  
</activity>  
  
<service android:name=".SimpleIME" android:label="@string/simple_ime"  
        android:permission="android.permission.BIND_INPUT_METHOD">
```

[see more](#)

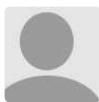
^ | v • Reply • Share ›



Deepak • 5 months ago

Works like a charm ! thank u sir.

^ | v • Reply • Share ›



Gisele Oliveira • 5 months ago

Great tutorial! How could do to create a button on the keyboard with the TextToSpeech function?

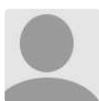
^ | v • Reply • Share ›



Nazz Patel • 5 months ago

using android studio 1.5. is dere necessary to create a folder xml.. it cant be done with layout files ?

^ | v • Reply • Share ›



Nauman • 5 months ago

i want to change asscci code to uni code on caps,but there is no asscci code for upper case letter.

Can you please help me.

^ | v • Reply • Share ›

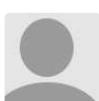


Nauman • 5 months ago

i want to change asscci code to uni code on caps,but there is no asscci code for upper case letter.

Can you please help me?

^ | v • Reply • Share ›



Nauman • 5 months ago

i want to change asscci code to uni code on caps,but there is no asscci code for upper case letter.

Can you please help me.

^ | v • Reply • Share ›



Carlos Gonzalez • 8 months ago

This works like a charm.

^ | v • Reply • Share ›



Marc Grant • 10 months ago

Does anyone know how we can set the newly created keyboard to become the default keyboard when the .apk installs itself?

^ | v • Reply • Share >



Igor Ganapolsky ➔ **Marc Grant** • 7 months ago

That's not possible. User will need to go into Language & Input system settings and choose the new keyboard.

^ | v • Reply • Share >



jeykeu • 10 months ago

How can I load a different layout when English(GB) is selected? Right now it just changes the flag on the spacebar based on what subtype is selected

^ | v • Reply • Share >



Tom • a year ago

How can I popup a mini keyboard overlay the soft keyboard by using TV remote control ?

^ | v • Reply • Share >



Igor Ganapolsky ➔ **Tom** • 7 months ago

Do you have an example of a mini keyboard online? I'd be curious to know what you mean...

^ | v • Reply • Share >



PrakashK Gowin • a year ago

I implemented this keyboard, but when I switch one edittext to another some of the keys gets hidden. Is anybody facing the same issue?? How to rectify this?

^ | v • Reply • Share >



witkola • a year ago

Using Android Studio 1.3, I got it to compile and I posted text to Facebook. I got the error message on my phone "Unfortunately, SimpleKeyboard has stopped". I've recompiled with no reported errors, uninstalled the first .apk and uploaded the latest but I still get the error message. That it worked the first time suggests that it may be related to the phone running 4.3 but I don't know if this is the case. Because there is no error reported in AS1.3, there is no stack flow info to indicate problems in the configuration. Has anyone else, using AS 1.3, run the keyboard successfully? If so, which Android version is running on the phone?

Update:

Using "CatLog" on my phone, I ran "Simple Keyboard" and got this log when it crashed:

"(mm-dd h:s) E/AndroidRuntime(8976): java.lang.RuntimeException: Unable to instantiate service com.example.(username).simplekeyboard.SimpleIME: java.lang.ClassNotFoundException: Didn't find class "com.example.(username).simplekeyboard.SimpleIME" on path: /data/app/com.example.(username).simplekeyboard-1.apk"

Where do I begin to correct this?

^ | v • Reply • Share >



Ariel Charles • a year ago

How to make an emoji keyboard? Please help

^ | v • Reply • Share >



tuts+

Teaching skills to millions worldwide.

22,264 Tutorials 876 Video Courses

Meet Envato +

Join our Community +

Help and Support +

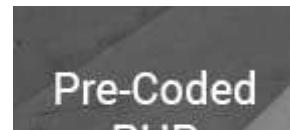
Email Newsletters

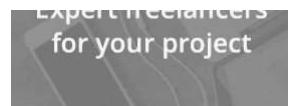
Get Envato Tuts+ updates, news, surveys & offers.

Email Address

[Subscribe](#)

[Privacy Policy](#)





Check out Envato
Studio's services

Browse PHP on
CodeCanyon

[Follow Envato Tuts+](#)

© 2016 Envato Pty Ltd. Trademarks and brands are the property of their respective owners.