GLOBILE

THE COMMON MOBILE PLATFORM

SanSecureKeyboard module Technical Documentation

Mobisec development

SanSecureKeyboard Module

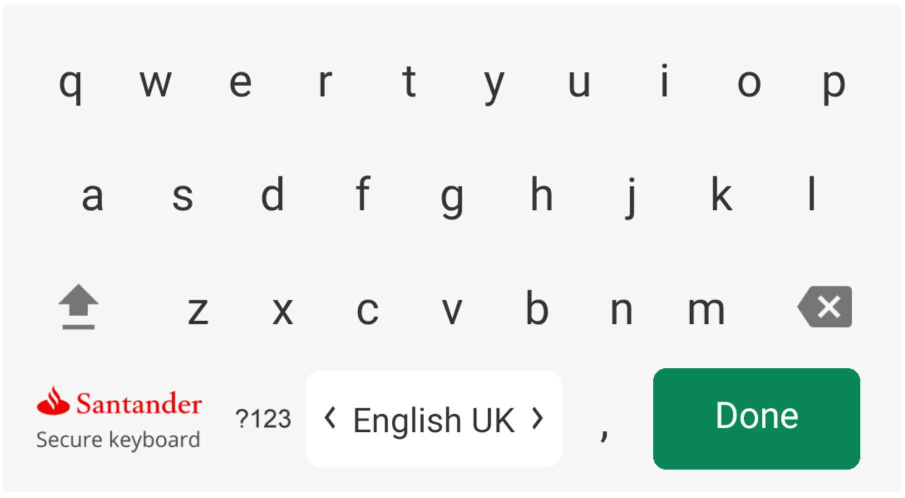
Module which offers a custom secure keyboard in order to interact with forms. This keyboard contains protection to tapjacking, and it works at in an independent way to S.O.

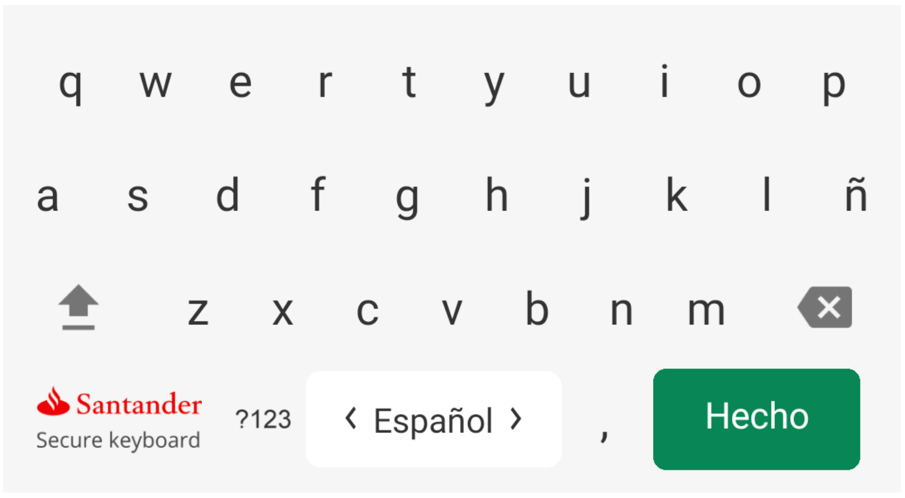
Repository

https://gitlab.alm.gsnetcloud.corp/mobi-sec/STGSanSecureKeyboard_Android

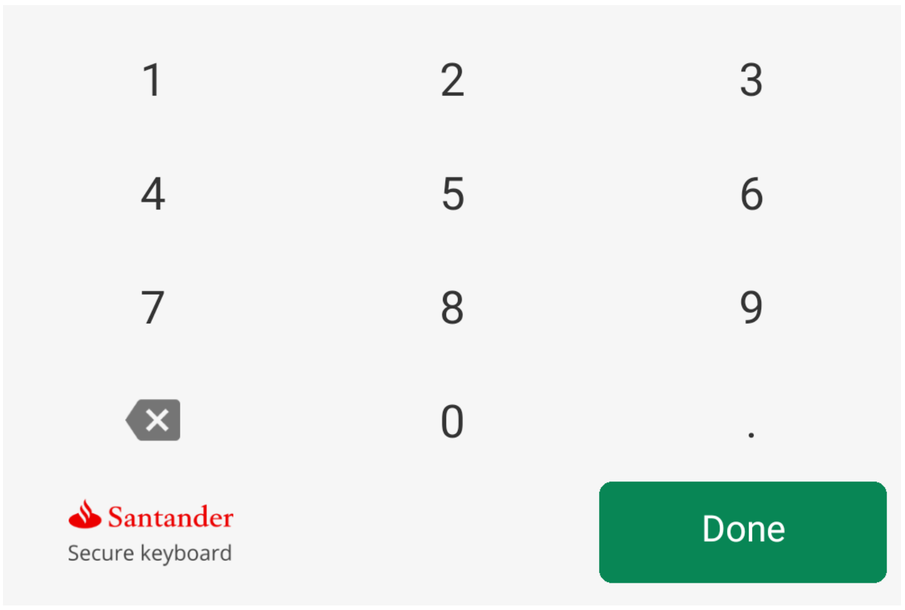
Types of Keyboard implemented:

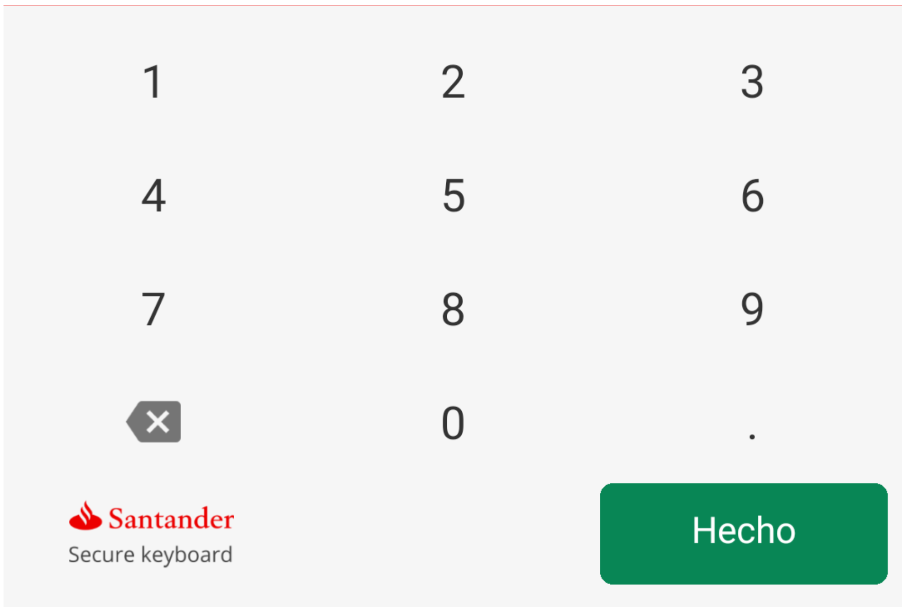
Alphanumeric



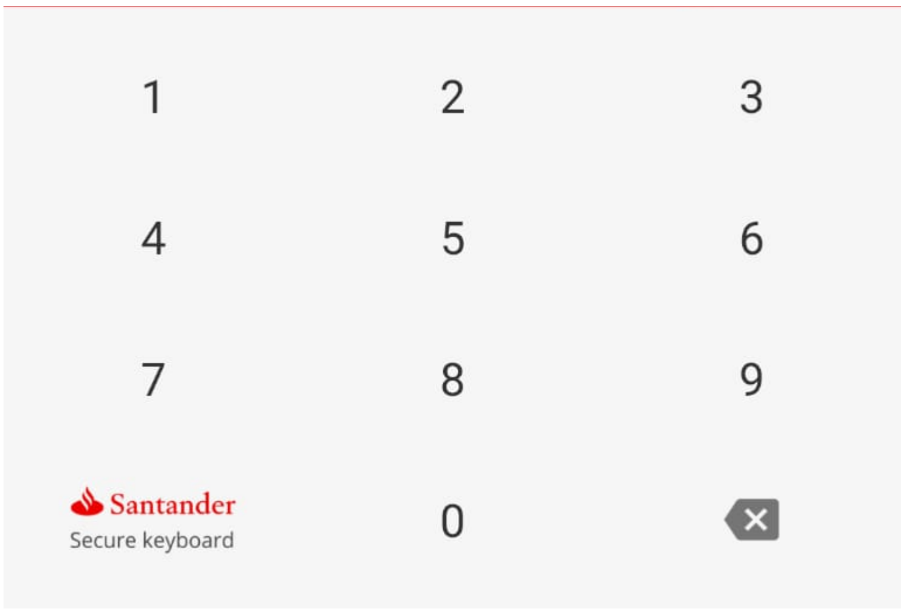


Decimal

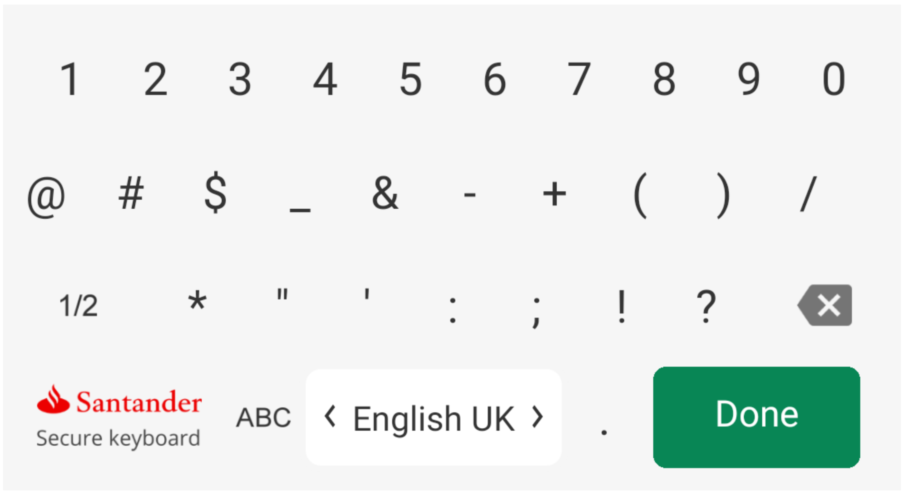

1
2
3
4
5
6
7
8
9
0
.
Santander
Secure keyboard
Done

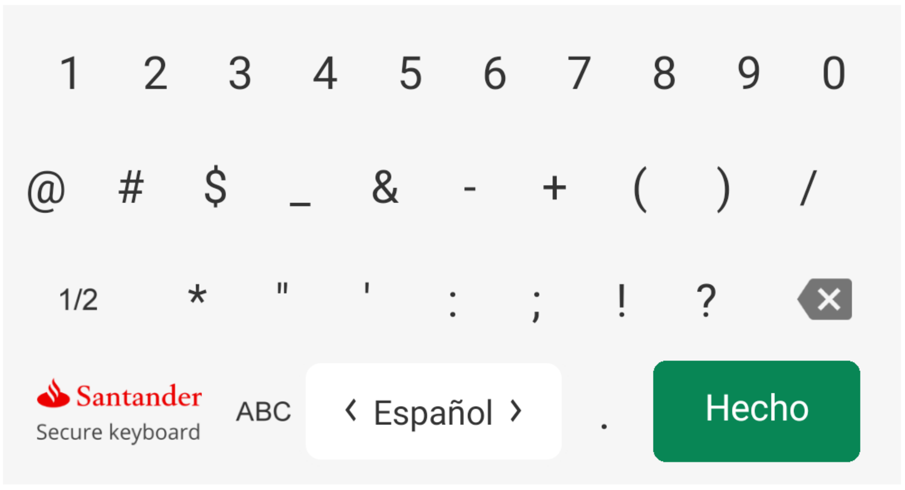

1
2
3
4
5
6
7
8
9
0
.
Santander
Secure keyboard
Hecho

Numeric

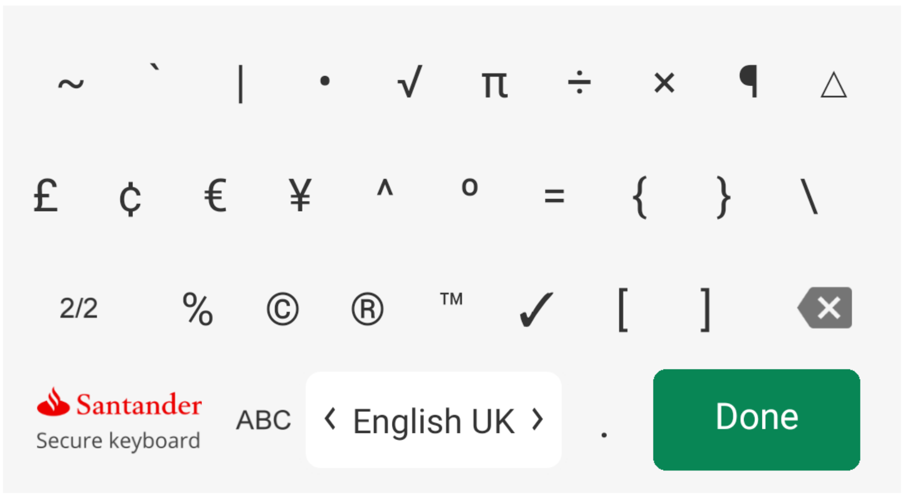


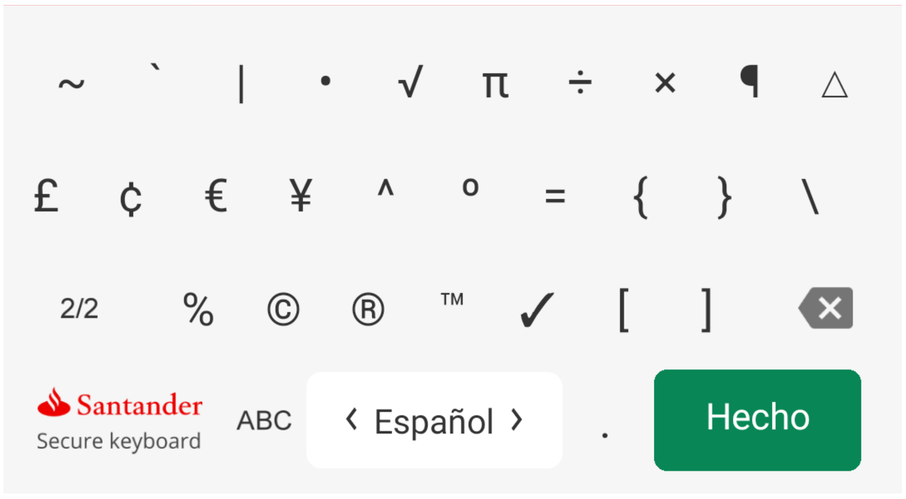
Special characters (1/2)



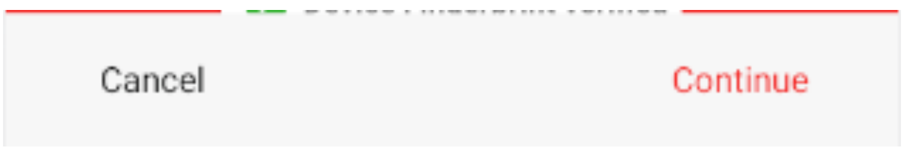


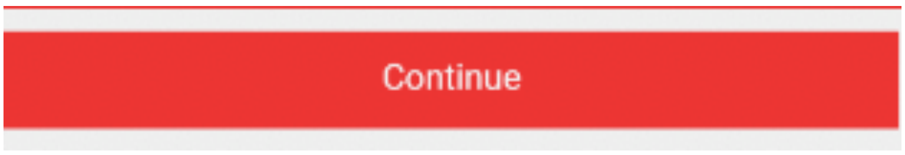
Special characters (2/2)





Top Row Buttons





Integration

To integrate SanSecureKeyboard module, the integrator must include their code in the application (manually) and then import the module in the Settings.gradle file:

```
include ':sansecurekeyboard'
```

Also, in the application build.gradle add this line in the dependencies:

```
implementation project(': sansecurekeyboard ')
```

Notes:

- 'sansecurekeyboard' is the folder where the code should be in the application folder. The developer may change this values.

The default implementation of SanSecureKeyboard module can be integrated using this sample code on a xml view:

```
<com.globile.santander.mobisec.securekeyboard.SanEditText
   android:id="@+id/login_password"
   android:layout_width="match_parent"
   android:layout_height="@dimen/editext_rounded_height"
   android:hint="@string/enter_pincode"
   android:inputType="numberPassword"
   android:privateImeOptions="nm"
   android:textColorHint="@color/ui_white_translucent"
   android:textColor="@color/ui_white"
   android:theme="@style/SanEditTextTheme"
   app:sanKeyboardType="numeric"
   app:sanKeyboardButtons="continueOnly"/>
```

As part of the customization, the integrator would change the keyboard type and add a top keyboard toolbar with "Cancel" and "Continue" buttons. The current available params are list below:

- **sanKeyboardType**
  - alpha – Alphanumeric keyboard.
  - alphaUpper – Alphanumeric keyboard with all upper letters.
  - alphaUpperPerm – Alphanumeric keyboard with all upper letters permanently.
  - decimal – Numeric keyboard with done button.
  - numeric – Simple numeric keyboard.

- **sanKeyboardButtons**
  - none - The upper row will not appear.

- continueOnly - A red “Continue” button will span the whole upper row.
- cancelContinue - Two buttons will appear in the upper row: “Cancel” and “Continue”.

Custom buttons

Currently, there are four custom buttons. These are defined in the SanKeyboard class.

- *ContinueOnly*: The continue button that spans all the top row.
- *Cancel*: The button at the left of the top row, when there are two.
- *Continue*: The button at the left of the top row, when there are two.
- *Done*: The button at the bottom right.

Their appearance is defined in the SanCustomKeyData inner class and are automatically created with the keyboard. They are language sensitive, as shown in the screenshots.

By default, all buttons are shown as enabled. To toggle the enabled status of the button, call the enableDisableCustomKey method of the SanEditText class.

Example:

```
loginPassword.enableDisableCustomKey(SanKeyboard.KEYCODE_CONTINUE, false);
```

The method will do nothing if the key passed to it is not a custom key.

Custom Actions

By default, when any custom button is clicked, the keyboard get hidden.

In order to add more actions, the developer may create a *SanKeyboardCallback* interface and set it to the SanEditText. The same callback may be set in various SanEditText but the idea is that every SanEditText should have a different callback.

The callback has two methods:

1. onContinueClick() → called when the continue button is clicked.
2. onCancelClick() → called when the cancel button is clicked.

The developer may use the abstract class SanKeyboardCallbackImpl and override one of the above methods.

Example:

```
private SanKeyboardView.SanKeyboardCallback loginPasswordKeyboardCallback
= new SanKeyboardView.SanKeyboardCallbackImpl() {
    @Override
    public void onContinueClick() {
        loginClick();
    }
};
```