Integrating Android applications with CommCare

> **This feature requires a CommCare Software Plan**
>
> This feature is only available to CommCare users with an Advanced Plan or higher. For more details, see the CommCare Software Plan page.

Overview

CommCare offers a few options for integrating with external applications, including providing a Content Provider for accessing case and fixture data and Broadcasts for detecting events like syncs and submissions. However, Intent Callouts are the most common and robust option for integrating applications as part of the form entry workflow. The goal of such integrations is usually to have the external application appear as seamlessly as if it were a native question type, with resultant data ending up in the CommCare session such that this data is manipulable and submittable by CommCare.

Implementation

Implementation of this integration takes place in two steps:

1. Definition of an "Android App Callout" question type in your CommCare form

2. Refactoring of your Android application to conform to the callout/response pattern defined here.

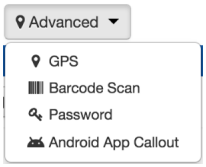
For a full discussion of implementing Android intents, see the Android Documentation.

App Callout Question

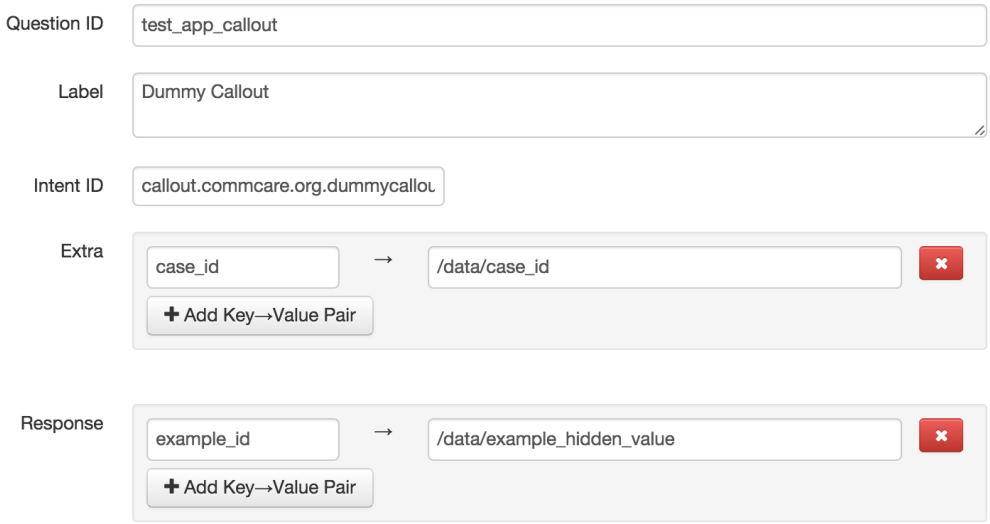
Assuming you have your external application installed and configured as above, you can use the form builder to create an Android Intent that will launch this application

Navigate to your form in CommCareHQ's form builder

Add an "Android App Callout" question from the "Advanced" menu in form builder



This will give you configuration options for the callout:



These fields are:

- **Question ID:** as usual
- **Intent ID:** intent action for your activity - for example "android.intent.action.DIAL" to launch the phone dialer
- **Extra:** keys and paths to form fields containing the values of arguments for the intent activity
  - The paths point to the **location of this data** - no hard-coding of values. The paths can point to hidden values that calculate the desired value
- **Response:** any extra values for ODK to store, along with the paths of where to store them

In this example we are calling out to an application filtering for the action name "callout.commcare.org.dummycallout" (explained below)

We are adding an Extra with the id "case_id" and the value taken from /data/case_id, a hidden value that we've set the value of previously (IE from case data, from another question, etc.)

We are receiving a Response with the id "example_id" that CommCare will load into the hidden value /data/example_hidden_value

Further, this question itself (test_app_callout) can *also* be set to a value as a result of the callout.

CommCareODK additionally supports customizing the button text and setting intent type by editing the underlying XForm XML and setting the following attributes on the odkx:intent element:

- button-label
- type

As an example, the following XML sets the intent type to "vnd.android-dir/mms-sms" and the button label to "Send SMS".

AndroidManifest.xml

```
<odkx:intent xmlns:odkx="http://opendatakit.org/xforms" id="send_sms" class="android.intent.action.VIEW" button-
label="Send SMS" type="vnd.android-dir/mms-sms">
        ...
</odkx:intent>
```

Now, let's get to building the corresponding application parts

Communicating with CommCare from your Application

For the source code referred to in this document please refer to this application.

First, in our AndroidManifest.xml we will need to register an intent-filter to listen for CommCare's call out. This will look like:

AndroidManifest.xml

```
<manifest>
...
    <application
        ...
        <activity
            android:name=".CalloutActivity"
            android:label="@string/app_name" >
            <intent-filter>
                    <action android:name="android.intent.action.MAIN" />
                    <action android:name="callout.commcare.org.dummycallout.LAUNCH" />
            </intent-filter>
        </activity>
        ...
    </application>
</manifest>
```

This tells Android that we want to listen for the Intent named *call.commcare.org.dummycallout*, as specified in the Intent ID of the form builder.

Next, in the activity's onCreate() method we parse the Extras from the intent bundle:

CalloutActivity.java

```
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);

                ...

        Bundle mBundle = getIntent().getExtras();
        //this is how we read in values sent by CommCare
        String caseid = mBundle.getString("case_id", null);
```

This will read in the extra defined in the form builder Extras block.

Finally, after this external application has done all of its work, we package up our results in a response. There are two ways of adding resultant data:

onClick

```
String text = entryBox.getText().toString();
Intent data = new Intent();

//this is how you would add responses to the default bundle.
Bundle responses = new Bundle();
responses.putString("example_id", "Example text");
data.putExtra("odk_intent_bundle", responses);

// this is the value that CommCare will use as the result of the intent question
data.putExtra("odk_intent_data", text);
setResult(Activity.RESULT_OK, data);
finish();
```

If you're only returning one string or integer as a result, then adding that as "odk_intent_data" is fine.

However, if you're returning multiple results, its generally best to put these in a response Bundle and have CommCare insert them into multiple hidden values.

Looking Forward

We will soon be adding the ability to callout to external applications with the syntax defined above from the Case List and Case Details screen. In the former case, you will be able to use the value in "odk_intent_data" to filter the case list. In the latter case, you will be able to use the data contained in the session within your external application.