Building Apps for CommCare Supply

Introduction

Before starting to build your Supply application there are a few concepts you must familiarize yourself with:

1. Locations (renamed Organizations)
2. Ledgers
3. Transactions
4. Repeat groups

In this document:

- Setting up iteration over the product list
- Saving date to a ledger: Setting up transaction questions
- Loading data from a ledger
- Setting up organizations for supply chain
- Supply-point case selection
- Summing data across all products
- Showing all products of a repeat group in one page
- How to refer to the product in a repeat? The use of current()

Setting up iteration over the product list

Why?

This will allow you to iterate over your product list to either read or write values for each of them.

How?

In order to iterate over a list of products, you need to set up a repeat group over the product list.

This is configured in the “products” repeat.

- Click on the repeat Advanced Model Iteration ID Query
- Click “Edit”
- Set the following:
  - Query Expression: `instance('products')/products/product/@id`
  - Instance ID: `products`
  - Instance URI: `jr://fixture/commtrack:products`
- Click “Save”
- This is the exact same for all product repeats.

Extra: If you want to run the product only over a defined list of products, you can use the Query Expression to filter the list. For example, the following expression will run the iteration over product of a defined program: `instance('products')/products/product[@program_id=xxx]/@id`

Saving date to a ledger: Setting up transaction questions

Why?

- This will allow you to create or update your ledger quantities.
- Used inside a repeat group, you can iterate over your product list and update for example the stock values.

How?

- Question ID: whatever you want
- Source/Destination case: this is the id of the case of the ledger or entity id.
  - If you want to apply the transaction to a ledger of the current case use: `instance('commcaresession')/session/data/case_id`
For supply chain, if you want to use the supply-point case of the user: instance ('commcaresession')/session/user/data/commtrack-supply-point
If you want to select the select supply point case (see here), use: instance ('commcaresession')/session/data/case_id_case_supply-point
If you want to save the ledger to a child case created by you form, use:
/data/subcase_0/case/@case_id  
data/subcase_0/case/@case_id or instance('commcaresession')/session/data/case_id_new_[case_type]_[index]
index should be 0 if you create one case.

- Any case id can be used here.
- Balance ID: This is the name of the ledger. The default should be “stock” but you can create any id you want.
  - Product: current()/../../../@id (3 dots)
    - This exact text should be used for a transfer inside a group.
    - If the transfer question is in a group in the repeat add one set of double dots and use: current()/../../../../@id
  - Quantity: value to set the ledger quantity with. It can reference a question in the repeat where you ask about, or calculate quantity. Normal calculates work fine here.

---

### Loading data from a ledger

**Why?**

- To read data previously saved into a ledger

Here is the calculate expression:

```
coalesce(instance('ledger')/ledgerdb/ledger[@entity-id=case_id]/section[@section-id='balance']/entry[@id=current()/../@id], 0)
```

```
instance('commcaresession')/session/data/case_id
```

Let’s break this down:

- coalesce([[big scary ledger reference]], 0)
  - just says use the ledger reference, and if you don’t find a value, use 0.
    - This is in case it is the first time filling in a form that touches that value.

Looking at the ledger reference

- instance('ledger')/ledgerdb/ledger[@entity-id=case_id]/section[@section-id='balance']/entry[@id=current()/../@id]
  - case_id: put here the id of the case where you want to get your ledger from. For example instance('commcaresession')/session/data/case_id
  - So instance('ledger')/ledgerdb/ledger[@entity-id=instance('commcaresession')/session/data/case_id]
  - Says grab the ledger for this case
- /section[@section-id='balance']
  - Says grab the section called “balance” (this corresponds to the “balance ID”). You could use here ‘stock’, ‘consumption’ ...
- /entry[@id=current()/../@id]
  - Says grab the entry value with the ID of the current product

This is pretty scary expression. Feel free to ask for help with it, though it shouldn’t vary too much per form.

---

### Setting up organizations for supply chain

Organizations can be used to do case sharing and therefore allow supply-point cases and ledgers to be shared between users.

Furthermore using organization levels will allow the configuration of different use cases such as stock management flow at different facility level or supervisor/field worker.

In the organization level page, after defining the structure of your organizations there are several options you need to set. Here they are:

- **Tracks Stock**: allow the organization to track stock, i.e. define ledgers.
- **Own Cases**: allow the organization to own the case data, i.e. all the cases saved by the user will be owned by the organization:
  - all users linked to the same location will be able to see the cases of this organization (same location case-sharing)
  - all users linked to a parent location, with the parameter “view child data” activated, will be able to see the cases of this organization (parent child case-sharing)
- **View Child Data**: allow the organization to see the case data of its child organizations

In the example below we have the following structure: departement > zs > commune > arrondissement > as > village_admin > village.
We follow stocks at 2 different levels:

- as linked to a supervisor
  - Tracks Stock: will allow the supervisors at this level to see the data of all its children in the village level
  - View Child Data: will allow the supervisors at this level to see the data of all its children in the village level
- village linked to a field worker:
  - Tracks Stock: will allow to track stock at village level.
  - Owns Cases: the village location will be the owner of the case data, allowing case sharing and parent locations to see the data.

With this setup the supervisor will be able to see his "child" user ledgers.

Supply-point case selection

Why?
Enable supervisor type of users to select any child supply-point to view and edit ledgers, or auto-select user own supply-point.

**How?**

First, you need to use the Advanced Module type. When creating your module, choose "New Advance Module"

- Your module settings > Case Management tab > Use the case type "supply-point"

![Module Settings](image)

- Your module settings > Product List tab > add the ledgers you want to display with the property "ledger:you_section_id"

![Product List](image)

- In your forms settings page > Case Management:
  - Click on "Add Action" + Add Action
  - Select "Load / Update / Close a case"
  - In the "Select case type" drop down, select "supply-point"
  - In the "Case list" drop down, select your module

![Forms Settings](image)
On the bottom of the page click on "Display the Product Stock for this Supply Point after it is selected"

Now when a user go to the module case list, all the supply-point at his level and underneath will be displayed.

The user can therefore select a supply-point and access to a form to edit the ledgers of that supply-point, by using this expression for a case id: `instance('commcaresession')/session/data/case_id_case_supply-point`

Let's consider this organization hierarchy. The supervisor is connected to the organization supply_cp while 2 field workers are connected to supply_rc and supply_rc1:

When connected to the application the supervisor, the case list will display all the relevant supply points.

Here the supervisor own supply point and the 2 field workers underneath him.
Summing data across all products

This is standard app-builder stuff, and can be done using the sum() function.

Outside the repeat group do a sum of a question inside the repeat group.

For example: sum(/data/products/item/data_product_payment) will sum the values of each product data_product_payment.

PS: the additional hidden word “item” in the path that is inserted due to the model iteration structure.

Showing all products of a repeat group in one page

When using repeat groups to iterate over product list, if you show a label or ask a question, each product will be shown in a separate page.

Depending on the number of products, displaying them all in one page can be more user friendly.

To do so just drag your repeat group in a question list:

How to refer to the product in a repeat? The use of current()

Referencing the ID of a product in the repeat can be tricky because it’s not obvious how many “..” expressions should be in the path.

This is a complicated question that involves understanding XML and XPath references, and particularly how they work in our tools.

However, you can use these general guidelines to help.

<table>
<thead>
<tr>
<th>Question you are referencing</th>
<th>Way to reference</th>
</tr>
</thead>
<tbody>
<tr>
<td>Display of the repeat group itself</td>
<td>current()/@id</td>
</tr>
<tr>
<td>Normal question directly in repeat</td>
<td>current()/../@id</td>
</tr>
</tbody>
</table>
The CommCare Supply transaction questions require special references. These are described below. These are all relative to a normal question in the same group, so additional `../` elements must be added to these paths when using the transaction questions inside groups.

<table>
<thead>
<tr>
<th>Question you are referencing</th>
<th>Way to reference (assuming no groups)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Product ID in a transfer question directly in repeat</td>
<td><code>current()/../id</code></td>
</tr>
<tr>
<td>Product ID in a balance question directly in repeat</td>
<td><code>current()/../id</code></td>
</tr>
<tr>
<td>For each additional group above the question</td>
<td>add an additional <code>../</code> to the path</td>
</tr>
</tbody>
</table>