

# Kitting Verification App

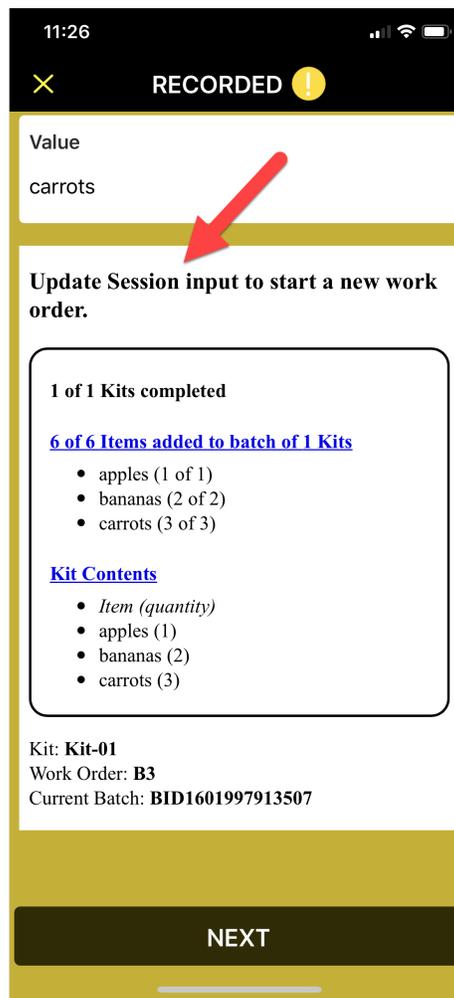
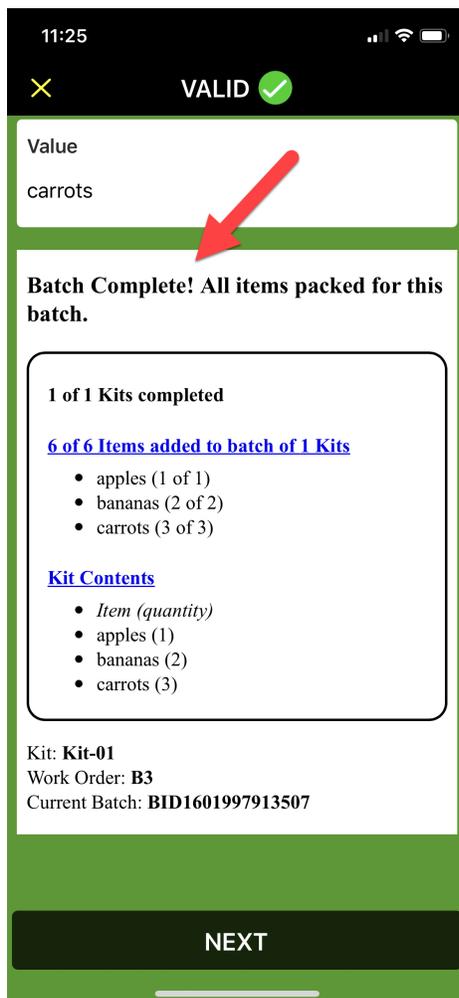
## The Workflow

The codeREADr "Kitting Verification App" works like this:

App users are given instructions from an administrator on which Work Order(s) to kit as well as the Kit-IDs and the items and quantities required for each kit.

When prompted by the app, an app user then enters one Work Order and one Kit ID into the app along with how many Kits are needed in total and how many will be done in each batch.

They then start adding items to a Kit by scanning each item and placing it into a bin. The app will track all scans and issue messages and alerts as needed to ensure accuracy.



## The Database

Here's a simple example of a kitting database uploaded as a CSV file to codeREADr's servers. We use fruit as the items so it's easier to understand.

Kit ID	Kit Contents
Kit-01	{"apples": 1, "bananas": 2, "carrots": 3}
Kit-02	{"apples": 1, "bananas": 2, "carrots": 3, "dates": 4}
Kit-03	{"apples": 1, "bananas": 2, "carrots": 3, "dates": 4, "eggplants": 5}
Kit-04	{"apples": 1, "bananas": 2, "carrots": 3, "dates": 4, "eggplants": 5, "figs": 6}
Kit-05	{"apples": 1, "bananas": 2, "carrots": 3, "dates": 4, "eggplants": 5, "figs": 6, "grapes": 7}
Kit-06	{"apples": 1, "bananas": 2, "carrots": 3, "dates": 4, "eggplants": 5, "figs": 6, "grapes": 7, "hominy": 8}
Kit-07	{"apples": 1, "bananas": 2, "carrots": 3, "dates": 4, "eggplants": 5, "figs": 6, "grapes": 7, "hominy": 8, "iceber_lettuce": 9}
Kit-08	{"apples": 1, "bananas": 2, "carrots": 3, "dates": 4, "eggplants": 5, "figs": 6, "grapes": 7, "hominy": 8, "iceber_lettuce": 9, "jicama": 10}

The **Kit ID** is in column 1 and the “contents” for each unique Kit in column 2. The format shown in column 2 is required.

Think of the **Kit Contents** like a Bill of Material (BOM) which includes all unique items for the Kit. After the item's colon, the quantity of that item is indicated for each Kit. So, for example, Kit-01 includes (1) apple, (2) bananas and (3) carrots to complete the Kit.

It's important to note that each scan creates a record that is synced to the codeREADr cloud but is also saved on-device to enable checking that the correct items are picked and that the correct quantities are picked.

Since the app is scanning offline, an individual kit and batch needs to be completed by the same device. However, Work Orders can be split between many devices providing the instructions are clear on which Kits each app user will be tasks with.

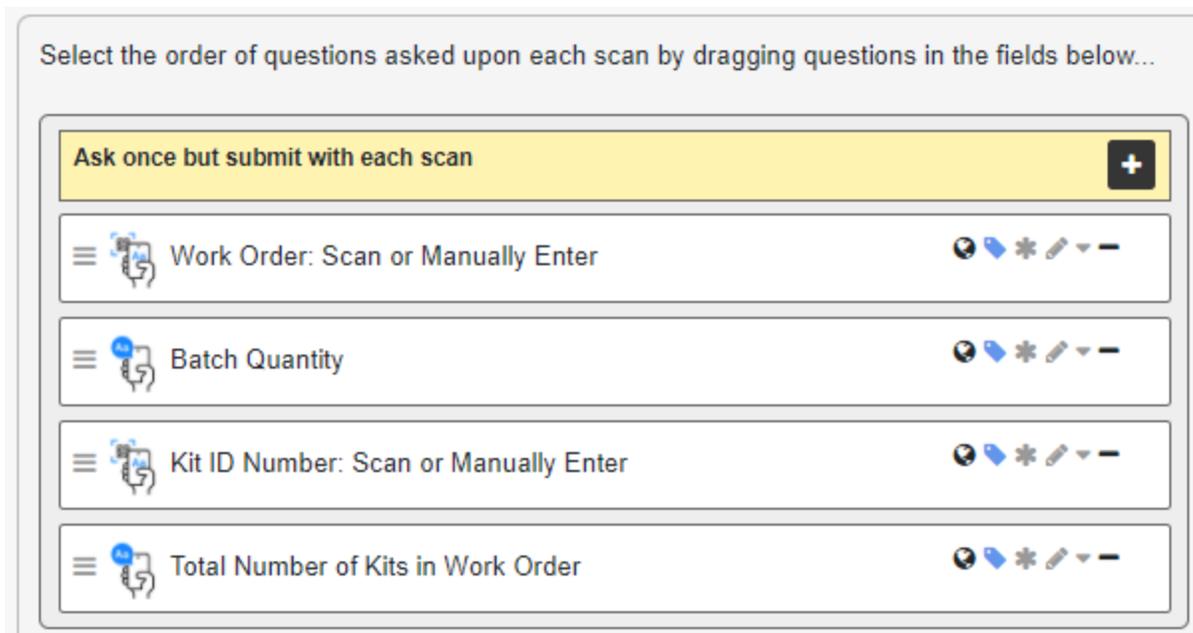
## The Service Type

You need to add a “**Validate Scans**” service type with **On-Device Validation** and **duplicates set to Valid**. You will need to associate that service with a database structured like the above example.

On the Advanced step you should enable Auto-Sync and Auto-Next Scan for Valid scans with an optional 3+ second delay.

## App User Prompts

In-app prompts ask the app user to answer questions. Note that the questions are placed under “Ask once but submit with each scan”. They are referred to as “session” questions because once answered they become part of the scan record until changed by the app user.



The app user will scan or enter the answers. When a Kit is completed, a new Kit ID needs to be entered. When all the Kits in a Work Order are completed, a new Work Order needs to be entered and the corresponding total number of Kits needed to fulfill that Work Order.

The Batch quantity can vary based on how many Kits can be bundled at the same time. So, for example, let's say a Work Order consists of 100 Kits. They can be done in 10 batches of 10 kits to fulfill the Work Order.

If the app user runs out of items or a schedule or quantity is changed, the app user can manually enter or scan an override code for the Work Order or Batch quantity. They can finish that Work Order or Batch at a later time and move on to another Work Order in the meantime.

Note: The Work Order, Batch Quantity and Kit ID questions are mandatory.

## Kitting Script

[The base kitting script is viewable and downloadable as a PDF here.](#) After editing it, you will need to copy and paste the script into the field under "Enable on-device custom validation" which you can find on the Advanced Step when configuring a service.

### 1. Question IDs

Notice that the script has a commented section for specifying the messages you want to give the app user. It also has a section to add the Question IDs you've added to the service. Note:

You can get each Question ID (“qid”) when hovering over the question or when on the website’s Questions tab.

Here’s an explanation of the question variables in the COV script relative to the above question examples:

var qidOrderId = “**Work Order: Scan or Manually Enter**”

[Here you put the Question ID where the app-user is prompted to enter the Work Order (or PO Number, etc.) as a session. It represents the total number of kits needed to fulfill the order.]

var qidBatchQty = “**Batch Quantity**”

[Here you put the Question ID where the app-user enters the Batch quantity as a session. The Batch quantity represents the number of kits to be kitted at the same time as a "Batch". An entire Work Order can be completed in one Batch or a Work Order can require multiple Batches.]

var qidKitId = “**Kit ID Number: Scan or Manually Enter**”

[Here you put the Question ID where the app-user is prompted to enter the Kit ID. The service’s database will have this value in column 1 of the associated database. In column 2 you will have the contents of the kit.]

var qidTotalKits = “**Total number of Kits in the Work Order**”

[Here you put the Question ID where the app-user is prompted to enter the total number of kits needed to complete the Batch (qidBatchQty) for the Work Order (qidOrderId).]

## 2. Overrides

It’s recommended to offer override codes for clearing in progress orders or batches. They can be scanned from the Tap to Scan screen or the Manual Entry screen. You can change the codes to whatever best suits your needs. The variables are:

```
var _resetBatchBarcode = "b123";  
var _resetOrderBarcode = "o123";
```

## 3. Messages

You can change the various messages presented to the app user based on specific events. They include:

```
var _msgItemAdded = "Item added.";  
var _msgItemBlocked = "Scanned item is not approved.";  
var _msgBatchComplete = "Batch Complete! All items packed for this batch.";
```

```
var _msgOrderComplete = "Order Complete! All kits packed for this order. Update the  
Session Info to start a new order.";  
var _msgWarnNewOrderNeeded = "Update Session input to start a new work order.";  
var _msgOrderReset = "You can start a new work order now.";  
var _msgBatchReset = "You can start a new batch now.";  
var _msgBatchResetFail = "There is no batch in progress to reset.";  
var _msgWarnBadSession = "There is an issue with your Session input.";  
var _msgWarnModSession = "Session input modified before completing order/batch.\nIf  
applicable, scan the reset order barcode or reset batch barcode to override.";
```

If you need help please contact [support@codeREADr.com](mailto:support@codeREADr.com).

##