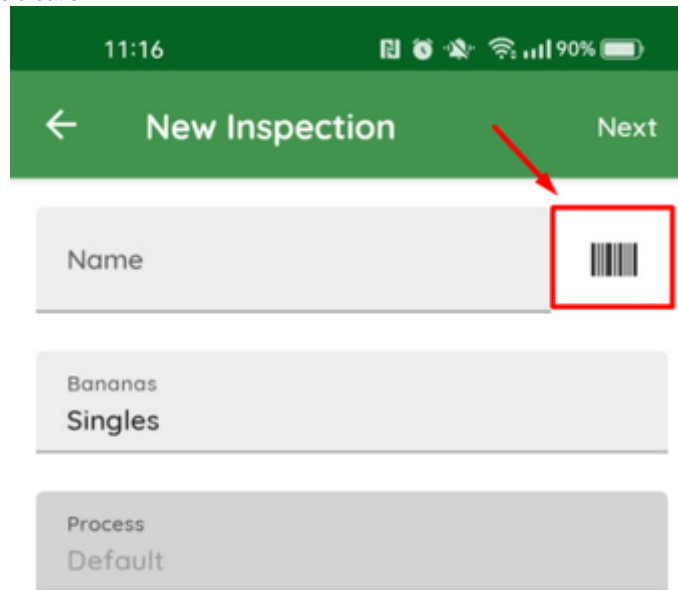# Integration - Barcode API - implementation steps

This integration allows an inspector to scan a barcode with the Clarifruit mobile application. The appropriate values will then be populated in the Clarifruit mobile application.

The barcode scanner will appear on the inspection's first page. It can retrieve information such as properties, standards, process, fruit & varieties, and by that - simplify the inspection's creation.

To implement this integration the Integration Partner is required to provide a barcode API for Clarifruit to invoke and field mapping settings within the Clarifruit application.

Overview

The barcode API is a single endpoint API (GET or POST) implemented by the Integration Partner, and invoked by Clarifruit.

The barcode API will be invoked by the Clarifruit server (which receives requests from the Clarifruit mobile application).

The API will receive an HTTP request from Clarifruit, and should return a flat JSON object with all relevant data (specified below).

Field Mapping

There are 5 Clarifruit entity fields to be mapped:

1. Variety (Optional)
2. Process (Optional)
3. Package (Optional)
4. Standard (Optional)
5. Properties (Optional)

Requirements

Following is an example of all the data required from a company for a barcode integration:

1. HTTP endpoint to get the barcode's information:
   a. The URL, e.g.: https://www.company.com/api/barcode
   b. The HTTP method (e.g. POST or GET)
   c. How the barcode's ID is passed to the server (can be either in the URL or in a template that can be sent in the POST body). For example:
      i. As a query string parameter:

```
https://www.company.com/api/barcode?id=123456
```

ii. As POST body:

```
curl -X POST https://www.company.com/api/barcode
    -H "Content-Type: application/json"
    -d '{"barcodeId": 123456 }'
```

2. Provide some examples of valid barcode values - that we could use for integration testing.
3. Decide which property will be triggering the API - inspection name / OR a barcode property.
4. The response's structure, with relevant mappings. Here's an example barcode return value from the company's barcode API:

```
5. {
    "CustomerFruitID": "Customer_Fruit_ID", // the company's fruit ID
    "CustomerPackageID": "some package id", // the company's package
    "CustomerProcessID": "some process id", // the company's process
    "CustomerStandardID": "some standard id", // the company's
standard
    "brand": "some value 1",
    "CountryOfOrg": "some value 2",
    "customsClrDt": 1676371031000
}
```

On Clarifruit's system, we would map the following:

- The IDs and values of the four Clarifruit entities/barcode fields (This way these entities will be known by Clarifruit's system):

```
CustomerFruitID, CustomerPackageID, CustomerProcessID,
CustomerStandardID
```

- The property field names (e.g. SomePropertyA) as well

Expected Values Format (in the JSON response):

Clarifruit Property Type	Expected Value format	Example
Text	String	"Abc123"
List	String	"Abc123"
Numeric	Integer / decimal	123 10.5
Date	timestamp - in milliseconds, or ISO format - ISO 8601.	1676371031000 "2022-12-08T07:22:02.369Z"

Mappings:

Special data types:

- The following data types will not be part of the mappings CSV, but we still need to map these:
  - **NameKey, Produce key, Variety Key, Process Key, Standard Key, Package Key**
    - You would need to tell us a key for each parameter that would be in the JSON response - to be associated & recognized by Clarifruit.
    - Clarifruit will map these accordingly.
  - You don't have to use all of them - for example, not all customers would like the barcode to auto-choose a standard & process, but most likely all the rest will be more relevant.