

# Unit 4 – Building a Barcode Scanner App

Certification: [C\\_LCNC\\_01](#)

Low-Code / No-Code Applications

# Barcode Scanner Application with SAP AppGyver

**Demo:** Barcode Scanner application

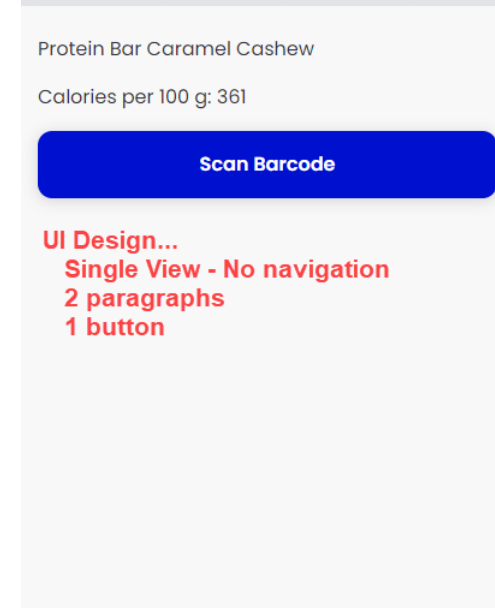
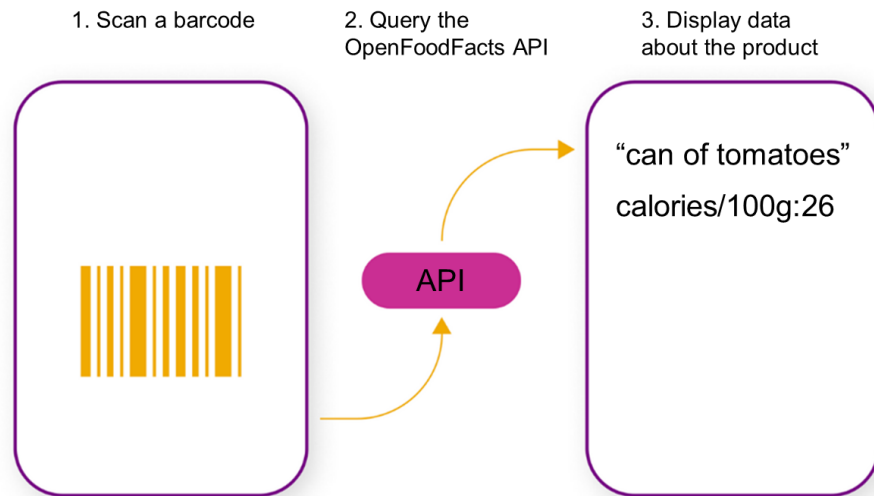


# Features

- Cross platform – Mobile (iOS, Android)
- Hardware integration
- Calling External API

# High level plan for UI, Logic and Data

- User taps button to scan barcode
- App uses barcode to query OpenFoodFacts API
- App displays 2 pieces of data about scanned product: name, calories



# Steps involved

1. Setting up the pre-requisites
2. Setting up the UI – Single View and Components
3. Logic for scanning barcode
4. Integrating with API
5. Displaying Food Product data

# Step 1 – Pre-Requisites

1. Subscribe to [SAP AppGyver](#) via a booster from SAP Business Technology Platform
2. Get the [SAP AppGyver Preview](#) app on your mobile phone

# Step 2 – View and its components

Protein Bar Caramel Cashew

Calories per 100 g: 361

**Scan Barcode**

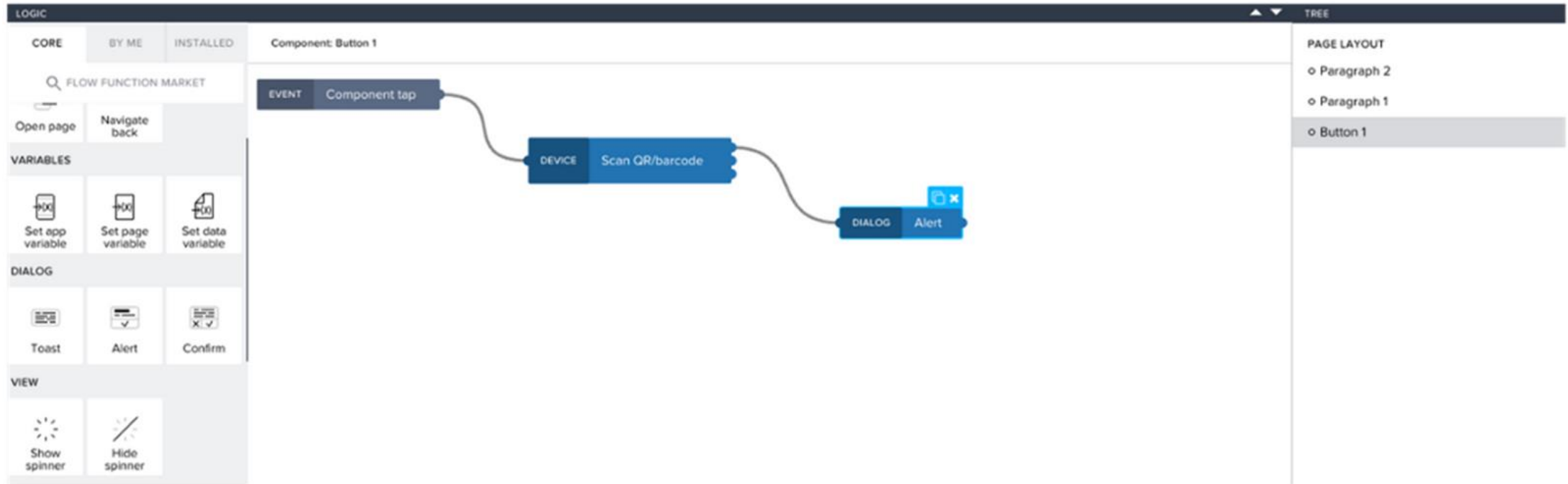
**UI Design...**

**Single View - No navigation**

**2 paragraphs**

**1 button**

# Step 3 – Logic for scanning barcode





# Step 4 – Integrating with an API

Publicly available OpenFoodFacts REST API

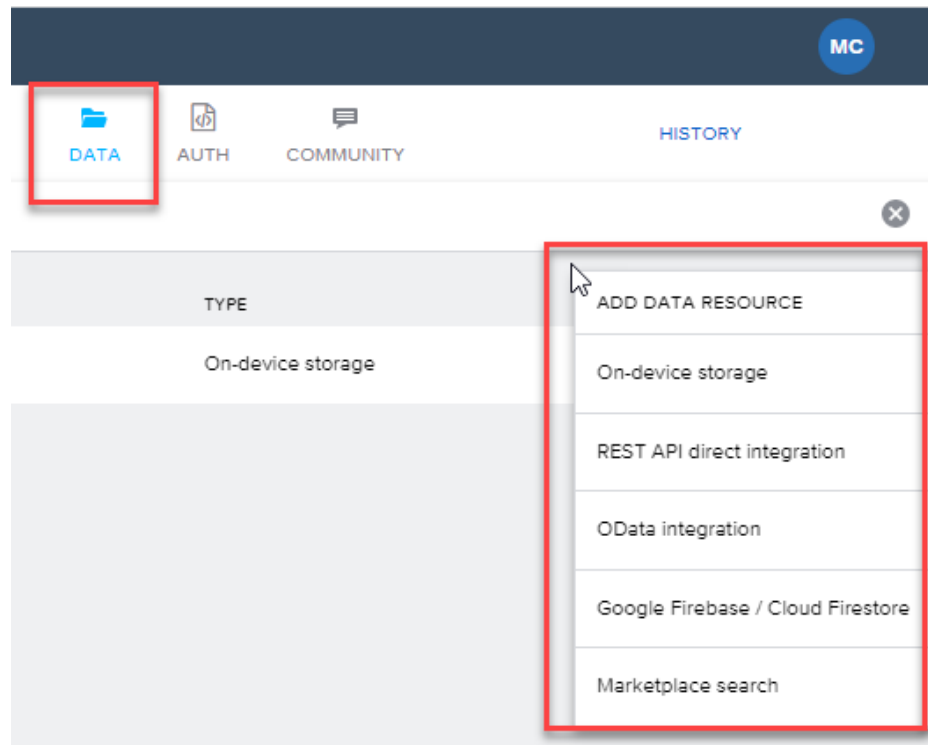
[https://world.openfoodfacts.org/api/v0/product/\[barcode\].json](https://world.openfoodfacts.org/api/v0/product/[barcode].json)

<https://world.openfoodfacts.org/api/v0/product/7350040162910.json>

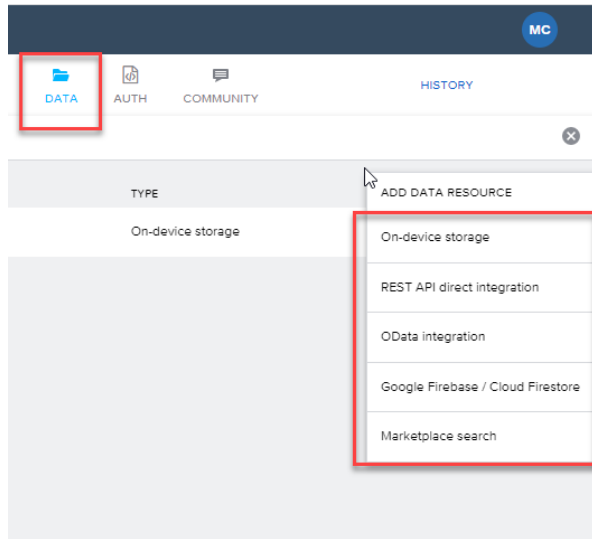
# Step 4 – Integrating with an API

## Data Resource

- Data that is stored outside the context of the running app
- **Data variable** is used to retrieve data from data resource



# Step 4 – Integrating with an API



REST API direct integration resource **openfoodfacts**

**Resource ID** ⓘ

openfoodfacts

**Short description** ⓘ

data from openfoodfacts api

**Resource URL** ⓘ

<https://world.openfoodfacts.org/api/v0>

**HTTP Header** ⓘ

No HTTP Header added. +

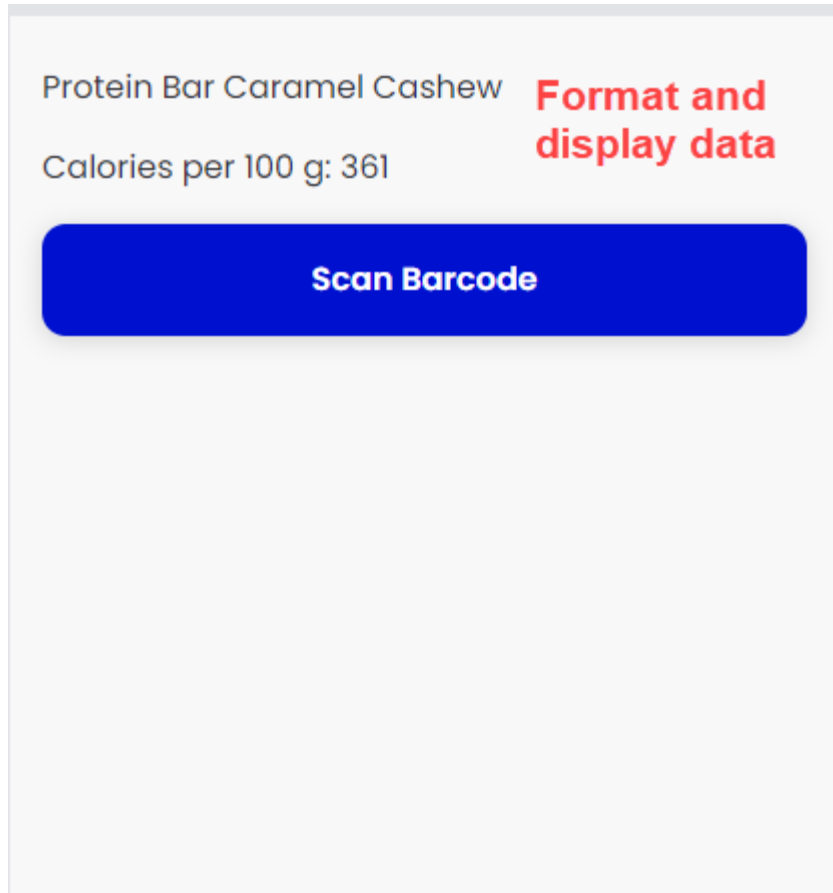
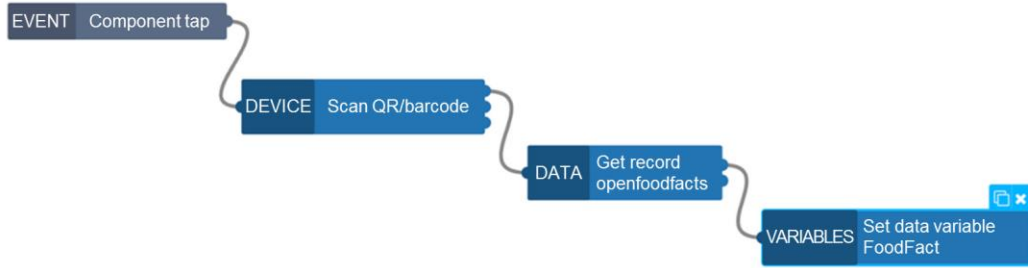
**URL placeholder** ⓘ

No URL placeholder added. +

**Query parameter** ⓘ

No Query parameter added. +

# Step 5 – Displaying Food Product data



# What have we learned so far

- Mobile application with SAP AppGyver
- High level plan
- Building the UI
- Building the logic
- Integrating hardware
- Calling an API

# Key Summary Points – Unit 4

## Question 1

*Choose the correct answer.*

What was the first real step you took in building your app with SAP AppGyver?

- ☐ Making a connection to OpenFoodFacts database.
- ✓ ☒ Making a high level plan that covered the user interface, logic and data.
- ☐ Mocking up the design and journey for the user experience.

# Key Summary Points – Unit 4

## Question 2

*Choose the correct answer(s).*

**What are the steps in the workflow that you planned for your SAP AppGyver app?**

- ✓ ☒ The user taps a button to scan a barcode.
- ✓ ☒ The app uses the barcode to query the OpenFoodFacts API.
- ☐ The app asks the user to provide their email address for authorization
- ✓ ☒ The app displays two pieces of data about the scanned product: food name and calories

# Key Summary Points – Unit 4

## Question 3

*Choose the correct answer(s).*

**What is true about the SAP AppGyver Preview app?**

- ✓ ☒ You can find the Preview app in all common app stores.
- ✓ ☒ The Preview app download is free.
- ✓ ☒ It's good to use the Preview app to see your changes in real-time.
- ☐ The preview app can be used with any no-code application development platform in the market.



# Key Summary Points – Unit 4

## Question 6

*Choose the correct answer.*

**What is a binding for?**




- ☐ Connecting to an API.
- ✓ ☒ Creating a link between a data variable and a UI component.
- ☐ Building the overall workflow.

# Key Summary Points – Unit 4

## Question 7

*Choose the correct answer(s).*

What are the parts you examined in the URL for the OpenFoodFacts API?

-  ☒ The version of the API.
-  ☒ The barcode number sequence for the product.
-  ☒ The JSON data format.
- ☐ The analytics tracking code

# Key Summary Points – Unit 4

## Question 8

*Choose the correct answer(s).*

**How did you create a data resource to be used in your SAP AppGyver app?**

- ✓ ☒ Opened the data tab from the top menu and selected “add data resource”.
- ✓ ☒ Selected “REST API direction integration” from the options.
- ✓ ☒ Entered the Resource ID for OpenFoodFacts.
- ☐ Used the entire URL provided by OpenFoodFacts.

# Key Summary Points – Unit 4

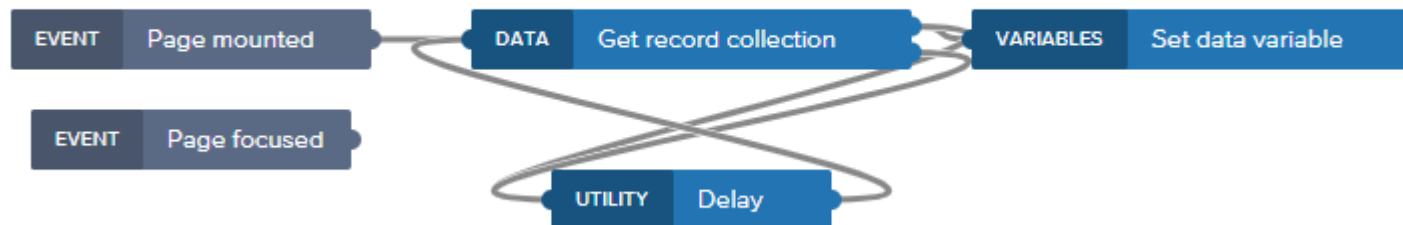
## Question 9

Choose the correct answer.

How can you reset the SAP AppGyver variable data logic in your logic canvas back to the default logic?

- ☐ Start over with building a new app.
- ☐ Clear the logic canvas and add a new logic flow.
- ☒ Select “reset data variable logic”.
- ☐ It's not possible – that's why detailed planning must be done in advance.

Data: notes



from data resource notesDB

Data variable name  
notes

Data variable type

- ☒ Collection of data records
- ☐ Single data record
- ☐ New data record

Filter condition

☒ No value

Ordering

☒ No value

Paging

☒ No value

RESET DATA VARIABLE LOGIC

Resets the logic canvas for this data variable to its default behaviour.

REMOVE FROM PAGE

# Key Summary Points – Unit 4

## Question 10

*Choose the correct answer.*

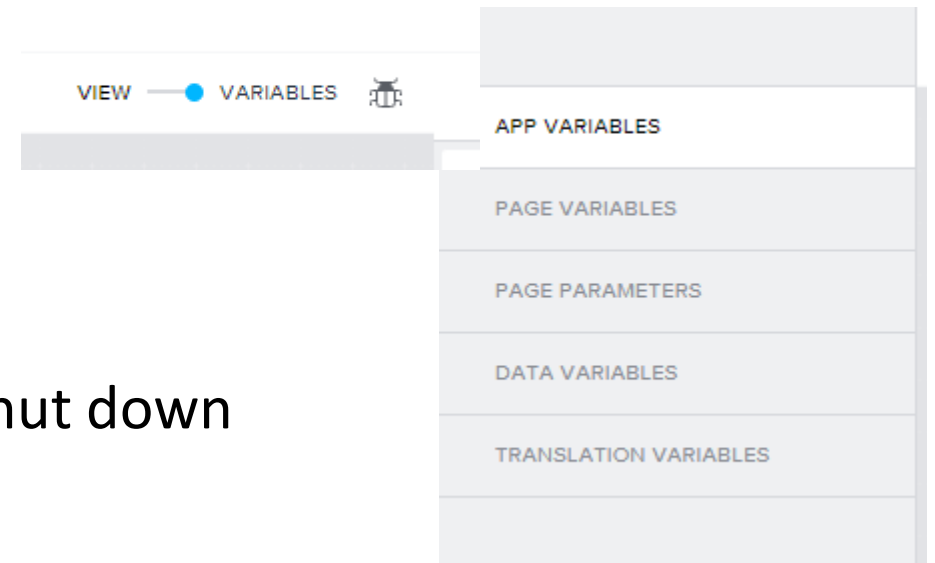
How did you edit the formulas to change the display of calorie data in the UI?

- ☒ Select the paragraph with “calories”, open the binding editor, choose the formula and edit.
- ☐ You need to be able to write in a programming language to do this.
- ☐ Added a new composable component with information on grams.

# Key Summary Points – Unit 4

## Variables

- **App Variables**
  - Exist globally
  - Initialized when app opens, exists until app is shut down
- **Page Variables**
  - Exists in the context of current page
  - Initialized when page opens, exists until page is closed
- **Data Variables**
  - Like page variables, used to read data from [data resources](#)



# Key Summary Points – Unit 4

## Data Resource

- Data that is stored outside the context of the running app
- **Data variable** is used to retrieve data from data resource

