Handbook for Developers and Consultants on SAP Build - Part 2

Deepak Kumar

Wilmington, USA
Email: *deepak3830[at]gmail.com*

Abstract: ***SAP Build offers a range of user - advanced tools to speed up app development and automation using smart AI technology. It lets users quickly build, enhance and automate apps, with drag - and - drop functions and AI - powered coding support. The platform makes it easy to expand applications by providing to - use connectors and business content for systems, not just SAP. Moreover, SAP Build promotes teamwork between business and development teams through integrated governance and lifecycle management features. Its main highlights include creating apps with both drag - and - drop methods and advanced coding techniques using AI assistance for coding tasks extending SAP and third - party apps rapidly and ensuring access, to SAP app data through pre - built authentication methods.***

Keywords: Low - code and no - code (LCNC), Citizen Development, SAP Build.

1. Introduction

Low - code and no - code (LCNC) refer to a new style of visual programming that enables application development without traditional programming. This approach significantly lowers the barrier to entry for anyone interested in developing applications.

Low - code combines traditional programming environments with no - code platforms, catering to developers with at least basic technical knowledge. It offers a hybrid approach, utilizing both coding and visual development tools.

No - code, on the other hand, replaces traditional programming languages with visual development tools, such as drag - and - drop components. This straightforward approach is accessible to both technical and non - technical individuals, including citizen developers.

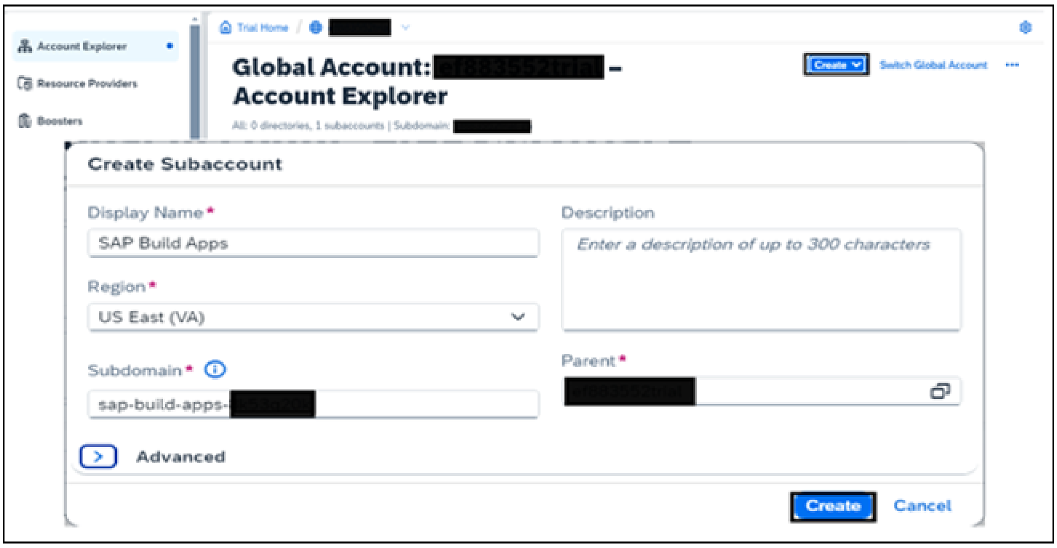
Citizen development addresses the current gap in technical expertise, particularly the shortage of experienced professional developers, by empowering non - technical users to create applications.

SAP Build's Low - Code portfolio merges no - code and low - code solutions, enabling users of all skill levels to easily create and enhance business applications. It allows business experts to build with drag - and - drop simplicity, integrate with SAP and non - SAP systems, and facilitate collaboration between business and development teams.

Pre - requisite: To understand this paper thoroughly prerequisite is the Handbook for Developers and Consultants on SAP Build - Part 1

Configuring SAP Build Apps in SAP BTP

To set up a subaccount, for working on SAP Build Apps begin by clicking the "Create" button in your account. Then select the "Create Subaccount" option, which will bring up a window. Enter a name for your subaccount; you'll see that the subdomain is created automatically. Pick a region from the choices provided. After entering all the required details hit "Create" to complete the setup.



Global Account: [redacted] – Account Explorer

All: 0 directories, 1 subaccounts | Subdomain: [redacted]

Create Subaccount

Display Name*: SAP Build Apps

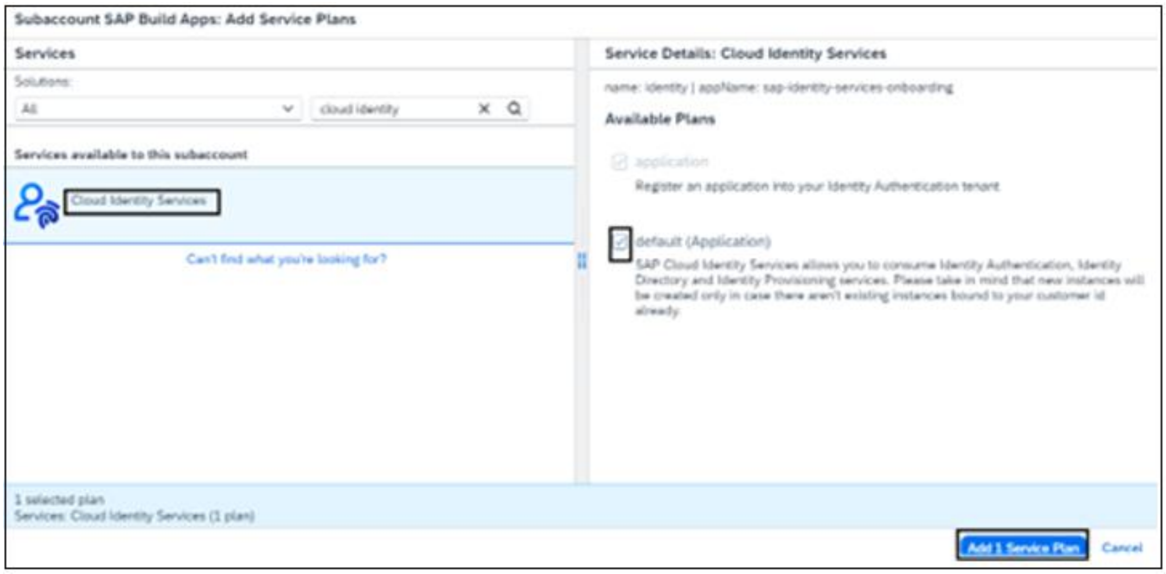
Description: Enter a description of up to 300 characters

Region*: US East (VA)

Subdomain*: sap-build-apps-[redacted]

Parent*: [redacted]

To improve user verification incorporate SAP Cloud Identity Services into your account. This feature when configured for a tenant can be applied to all subsidiary accounts. Start by establishing the Cloud Identity Service for each account. Initially go to the "Entitlements" section. Select "Entity Assignment. " Pick the subaccount then click on "Edit" and opt for "Add Service Plans. " Look up Cloud Identity Services. Choose the plan, for user management and verification. This guarantees safe user authentication throughout your subsidiary accounts.



Subaccount SAP Build Apps: Add Service Plans

Services

Solutions:

All

cloud identity

Services available to this subaccount

Cloud Identity Services

Can't find what you're looking for?

Service Details: Cloud Identity Services

name: identity | appName: sap-identity-services-onboarding

Available Plans

☑ application

Register an application into your Identity Authentication tenant

☑ default (Application)

SAP Cloud Identity Services allows you to consume Identity Authentication, Identity Directory and Identity Provisioning services. Please take in mind that new instances will be created only in case there aren't existing instances bound to your customer id already

1 selected plan

Services: Cloud Identity Services (1 plan)

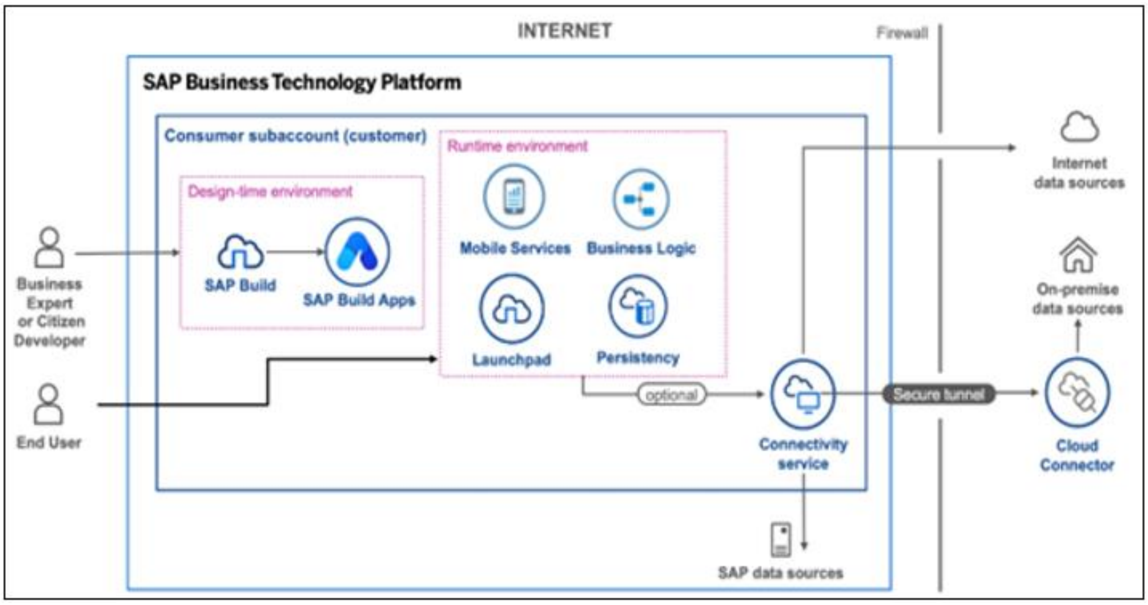
Add 1 Service Plan

Cancel

Furthermore, you should enable the entitlement, for SAP Build Apps. Choose between the free service tier based on your specific needs.

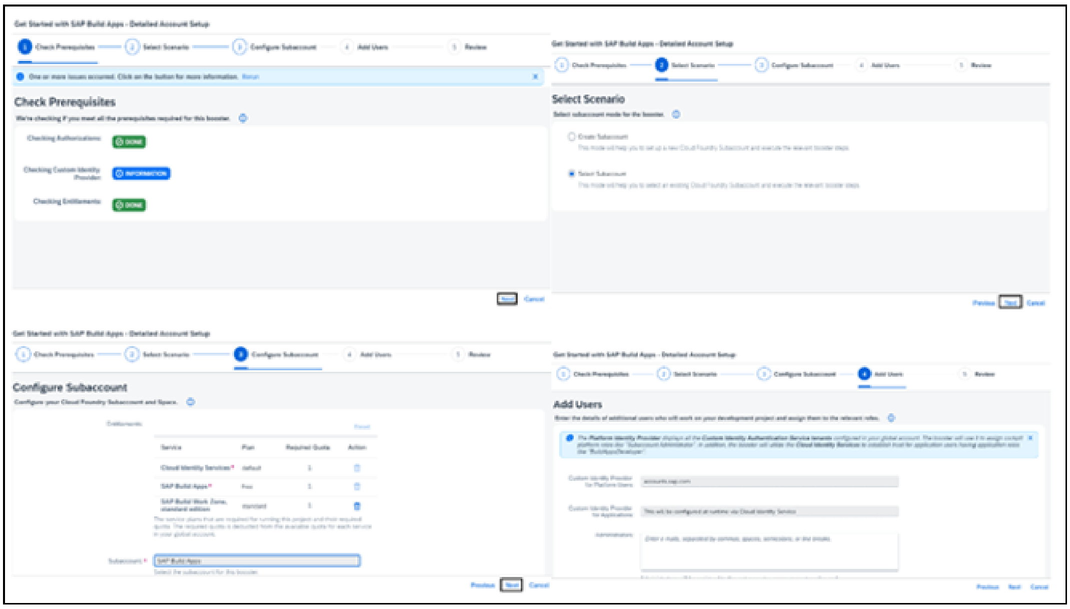
Once you have assigned the entitlements proceed to set up the services using either a booster or manual configuration method.

To subscribe to these services go to your account opt for "Booster " and search for "SAP Build Apps. " Utilize the booster tool to configure SAP Build Apps. Click on the booster option, which will conduct checks like verifying entitlements. Next decide whether to use an existing subaccount or create an one depending on your requirements. Customize the subaccount and services as, per your situation. Include users and administrators for your service carefully review all details. Then select "Finish. "



To sign up for the services go to your account click on "Booster " and look for "SAP Build Apps. " Use the booster to set up SAP Build Apps. Open the booster to run some checks like confirming the permissions. Next decide if you want to use an existing subaccount or create one based on your requirements. Customize the subaccount. Select the services you wish to include in your setup. Add users and admins for your service check all the details and then hit "Finish. "

Paper ID: SR24724163944 DOI: https://dx.doi.org/10.21275/SR24724163944



Get Started with SAP Build Apps - Detailed Account Setup

1 Check Prerequisites — 2 Select Scenario — 3 Configure Subaccount — 4 Add Users — 5 Review

One or more issues occurred. Click on the button for more information.

Check Prerequisites

We're checking if you meet all the prerequisites required for this booster.

Checking Authorizations	DONE
Checking Custom Identity Provider	INFORMATION
Checking Entitlements	DONE

Next Cancel

Select Scenario

Select subaccount mode for the booster.

○ Create Subaccount

● Select Subaccount

Previous Next Cancel

Configure Subaccount

Configure your Cloud Foundry Subaccount and Space.

Entitlements

Service	Plan	Required Quota	Action
Cloud Identity Services*	default	1	🗑
SAP Build Apps*	free	1	🗑
SAP Build Work Zone, standard edition	standard	1	🗑

Subaccount*: SAP Build Apps

Select the subaccount for the booster.

Previous Next Cancel

Add Users

Enter the details of additional users who will work on your development project and assign them to the relevant roles.

Custom Identity Provider for Platform Users: accounts.sap.com

Custom Identity Provider for Applications: This will be configured at runtime via Cloud Identity Service

Administrators:

Previous Next Cancel

Developing applications using SAP Build Apps

SAP Build Apps, formerly known as SAP AppGyver is a user tool designed for creating enterprise applications without the need, for coding. This platform allows users to build apps by dragging and dropping made components and customizing them to fit their requirements. It facilitates integration with backend data sources like REST and OData services, including SAP systems through a straightforward interface that enables data presentation and utilization in calculations. Additionally, users can define application logic by utilizing drag - and - drop flow functions to determine how the app reacts to user interactions or events.

The app development process follows a methodology. Includes essential components such as the User Interface (UI) which represents what users see on the screen and interact with; Data, which comprises information fetched from backend systems typically displayed in the UI; and Logic, which dictates actions taken in response to user interactions or application events.

During app development, initial emphasis is often placed on designing the layout of the app—focusing on how it appears and presenting desired information, to users before considering where or how that information is sourced. Creating an app involves designing its layout, including elements, like text boxes, input fields, pages, navigation menus, buttons, and other components that users interact with.

In some cases your app will need to store data such as user - generated content or preferences. Regardless of where the data comes from you'll have to use variables to hold it. Interactivity plays a role in apps requiring logic to respond to user actions or events within the application. To make the app functional you'll set up data resources to connect to data sources or APIs by specifying connection details like URLs, HTTP headers, and query parameters. The retrieved data is typically stored in data variables.

For apps connected to SAP Business Technology Platform (BTP), you often need to establish connections with SAP systems such as SuccessFactors or S/4HANA. This can be achieved through destinations—connection details stored in SAP—to streamline authentication processes for back - end systems. These destinations are created by SAP BTP administrators for application developers' use.

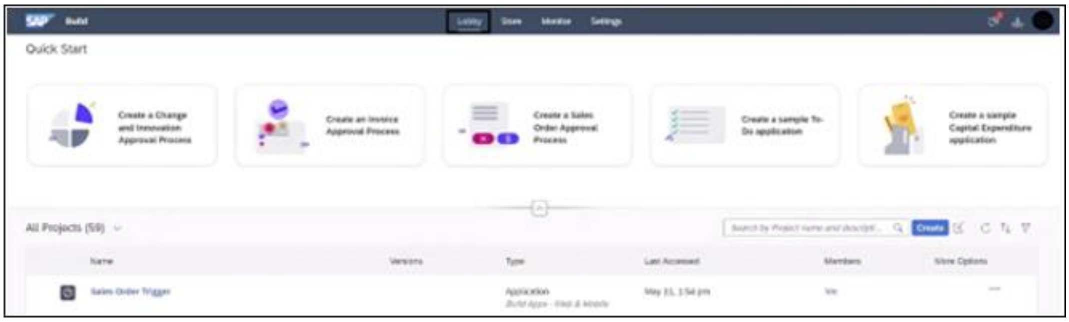
In addition to storing data in variables and fetching information from sources there may be instances where you need your own data storage solution. While setting up your SAP HANA database or another type of datastore is an option SAP Build Apps offers an approach, through Visual Cloud Functions (SAP Build Apps backend) for this purpose.

Visual Cloud Functions enable you to specify entities, such, as database tables for storing information that can be accessed by all users of your application. Additionally, they empower you to define server - side functions that boost efficiency and maintain secure logic. The data stored within Visual Cloud Functions remains intact when the application is inactive.

To make your application accessible, to users, you will need to deploy it on SAP BTP, which generates an HTML - 5 based application.

Familiarizing with Lobby:

Paper ID: SR24724163944 DOI: https://dx.doi.org/10.21275/SR24724163944



Quick Start

Create a Change and Innovation Approval Process

Create an Invoice Approval Process

Create a Sales Order Approval Process

Create a sample To-Do application

Create a sample Capital Expenditure application

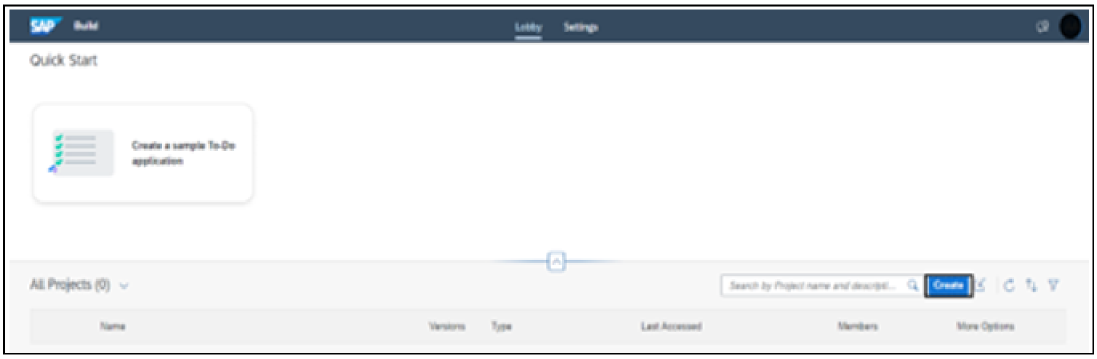
All Projects (59)

Search by Project name and descripti...

Name	Versions	Type	Last Accessed	Members	More Options
Sales Order Trigger		Application Build Apps - Web & Mobile	May 31, 1:54 pm	Me	···

The **lobby** is where the three SAP Build tools - SAP Build Apps, SAP Build Process Automation, and SAP Build Process Work Zone come together most closely. Here, you can see SAP build projects and create new projects of various types. In the lobby, you can sort and filter the projects. Your team may have created hundreds of projects, but you want to find the one you are working on quickly. You can filter projects by a keyword in the title or description of the projects. You can open the filter controls and filter by, for example, project owner or project type. For any project, there are additional actions you can take. The most common actions are managing who can work on the project, importing or exporting the project, renaming the project, and duplicating the project.

Creating a project: Choose Create as highlighted in the below screen. In the popup that appears on the next screen, you will see three options Build an Application, Build an Automated process, and Build a business site. You have to choose the Build an application option as shown in the below screenshot. Then will see options such as web and mobile application and application backend for this we are choosing the web and mobile application to build a front - end application. In the next popup, you will be asked to provide a project name and short description choose a project name add a description, and click on Create.



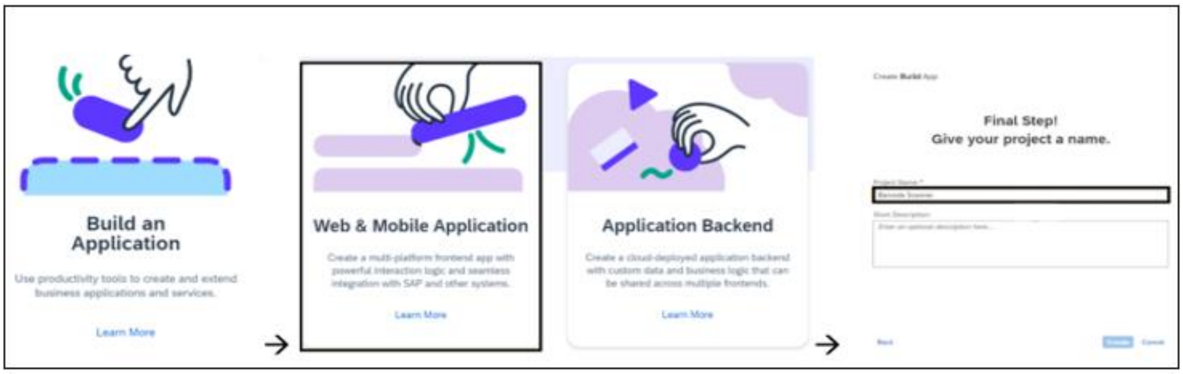
Quick Start

Create a sample To-Do application

All Projects (0)

Search by Project name and descripti...

Name	Versions	Type	Last Accessed	Members	More Options



Build an Application

Use productivity tools to create and extend business applications and services.

Web & Mobile Application

Create a multi-platform frontend app with powerful interaction logic and seamless integration with SAP and other systems.

Application Backend

Create a cloud-deployed application backend with custom data and business logic that can be shared across multiple frontends.

Create Build App

Final Step!
Give your project a name.

Project Name *: Barcode Scanner

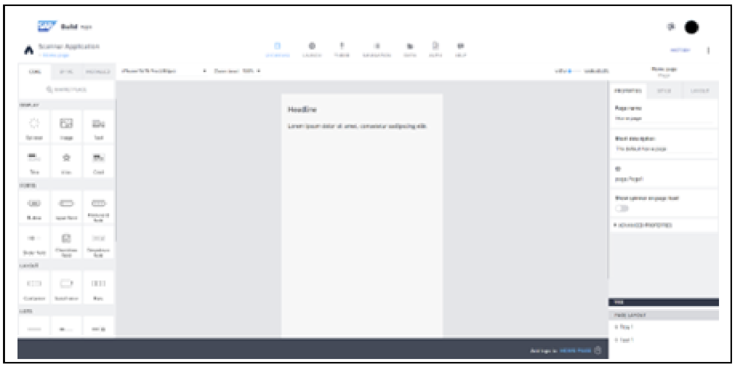
Short Description: Enter an optional description here...

Using the Composer for building:

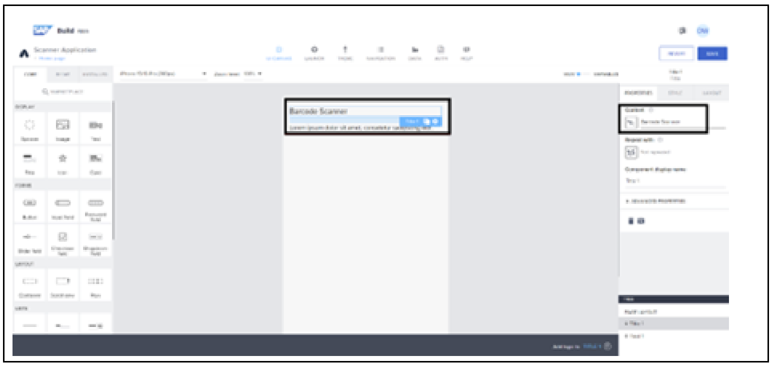
Here's a glimpse of the app's home page, with the logic panel at the bottom hidden. The Page Selector allows you to create pages and switch between them. Special Areas, which are frequently used but crucial include Data for importing data into the app and Launch for testing it out. The Canvas is where you design the user interface that users will interact with. Components consist of elements like buttons and images that can be added to the Canvas to build your app.

Properties is where you customize these components and other settings related to pages or logic elements. In cases where your page has components organized in hierarchies making it challenging to manage their relationships or select one individually all components are displayed in a tree hierarchy, in the Tree View section. Additionally, there is a pane where you define how the app should respond to user actions and events. By default, this pane is closed. Can be opened by clicking a link at the bottom of the canvas.

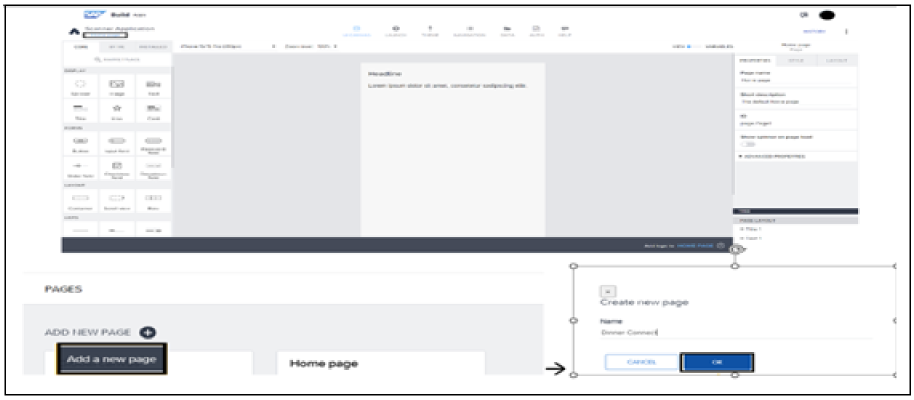
Paper ID: SR24724163944 DOI: https://dx.doi.org/10.21275/SR24724163944



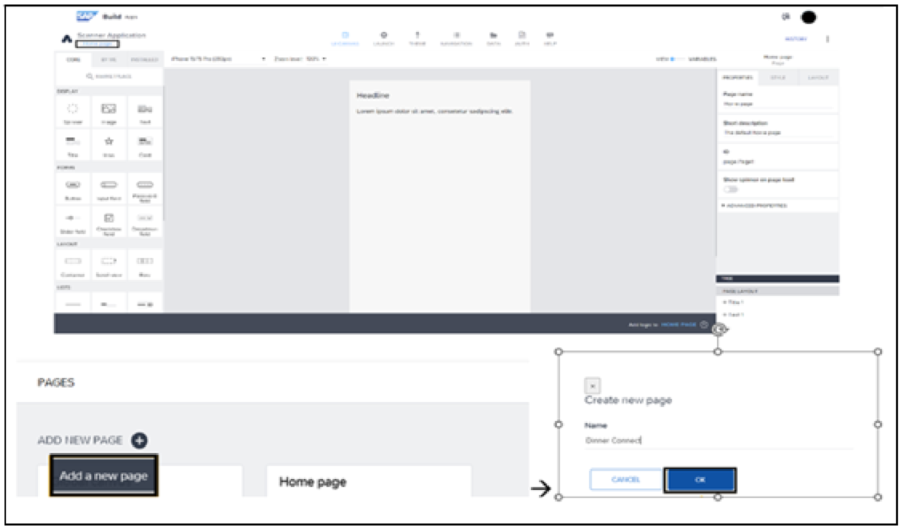
Editing content: to edit the content of the application you have to click on the content you want to edit and then on the right side with editing tools you can edit the content text. And use properties according to your need such as several lines making it selectable content etc.



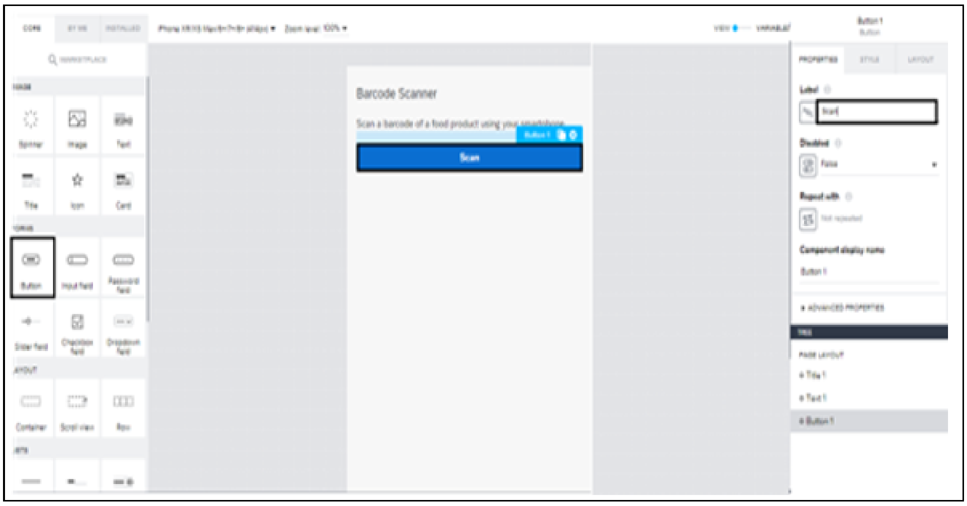
Adding a page: While adding a new page click on the home page as highlighted in the below screenshot. It will open another tab where you can see multiple pages there you need to click on "Add a new page" button and then give a name to the new page then click on the ok button to proceed. If you want to add multiple pages to your application, you have to repeat the above steps. After you are done adding click on the save button on the top right of the screen.



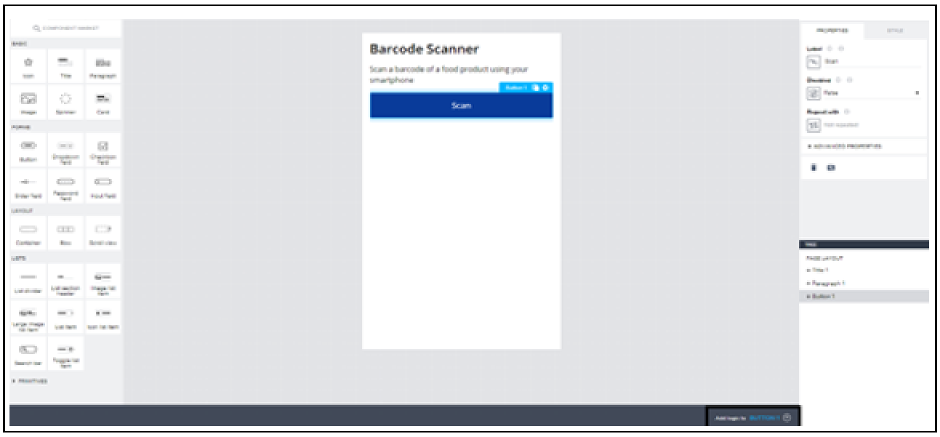
Modifying navigation to the pages: In the header of the composer application, you can see the navigation button by clicking on it. There you can enable or disable the navigation header bar by clicking on the highlighted area in the middle and then on the right side, you can see the properties appear. For the second part in the navigation menu, you can add the navigation to different pages by adding the item as such repeating the process of as many pages as you want. After you are done adding click on the save button on the top right of the screen.



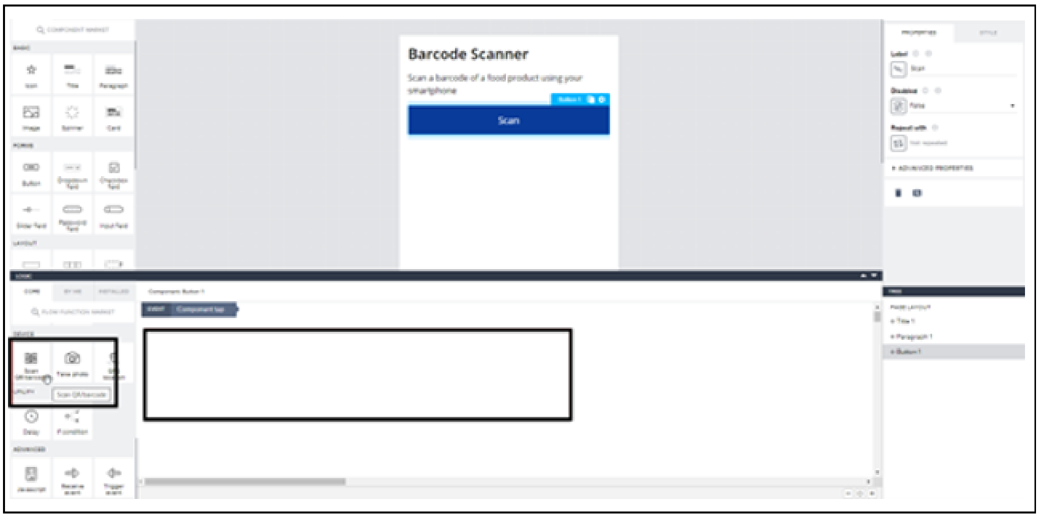
Adding a button: To add a button you have to go to components in the left part of the screen and click on the button as highlighted below. The button will appear on the screen as highlighted in the middle of the screen and you can edit the properties of the button on the right part of the screen in the properties.



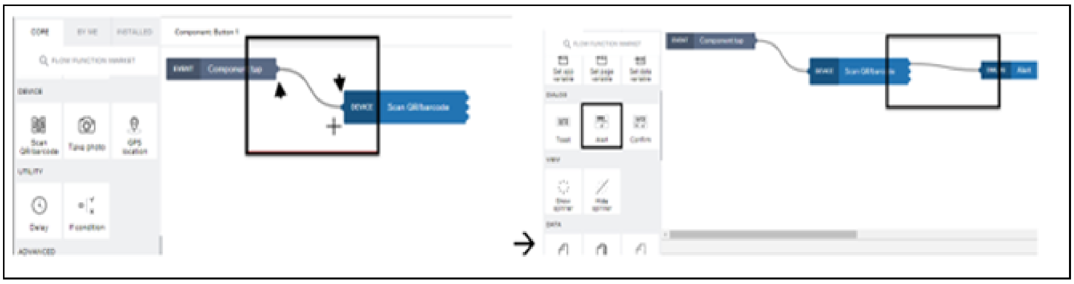
Adding logic to the button: To add logic to the button click on the add logic to the Button1 to open the logic pane to add logic.



In the logic pane, you can see the multiple logic available such as barcode, using the camera to take a photo, and GPS location. Delay, if condition, JavaScript, receive, or trigger event are a few of the core logics present, you can look for more in the marketplace or check the installed ones for your custom requirements in the left side of the screen. In our case, we are going to add scanning functionality. As you can see, the event for the button is “Component tap” here and the rest is empty. After clicking on the Scan/QR barcode, it will show in the logic pane.

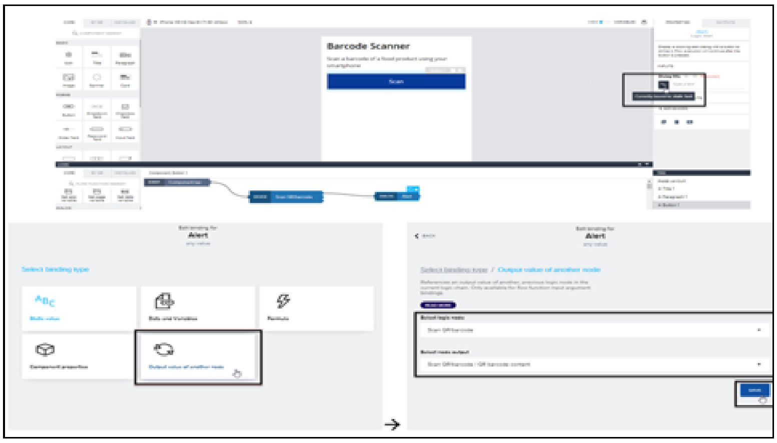


Click on the existing Component Tap connection. Manually create a link, to the Device connector. This connection serves to show the direction of flow which in this scenario involves tapping a button and then triggering the camera device to open. To test if the QR/barcode scanner is functioning properly you will incorporate a component into the application. This will display a popup alert once a barcode is scanned, showing the barcode number on screen. To accomplish this go back to the logic options locate Alert in the list and drag it next to the Scan QR/barcode logic. Connect them with a line between the Scan QR/Barcode option and the Alert element to indicate flow direction. Refer to the screenshot, for guidance.



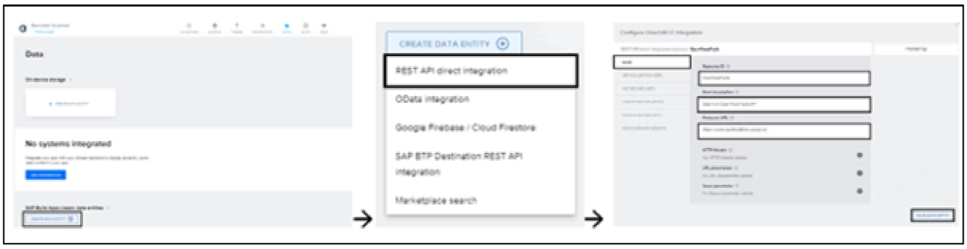
The alert currently shows a fixed text response. We aim to make it more dynamic by adding a binding. This will allow us to customize the information displayed on the barcode that has been scanned. To achieve this go to the element access the properties panel and select "Currently bound to; text" to open the binding options screen. Choose "Output value of another node "and set up the binding as follows; Select logic node – Scan QR/barcode Choose node output – Scan QR/barcode / QR barcode content. Finally, click Save.

Connecting the application to an API: Open your draft application in the Composer account.

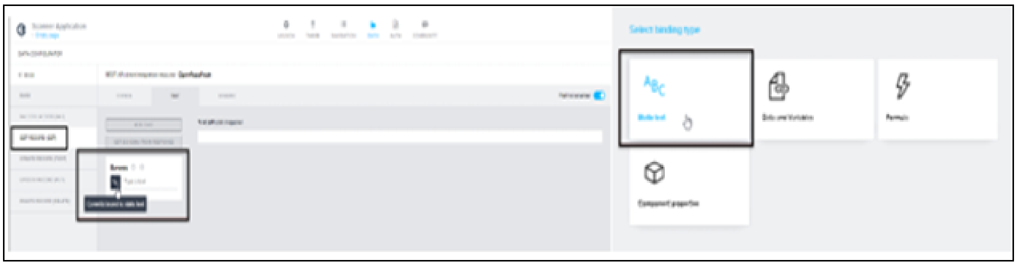


Click your Scan button, and then click the Data tab**.** Add a connection to a REST API. If you are using SAP Build Apps, then click SAP Build Apps classic data entities > Create Data Entity, and then select REST API direct integration.

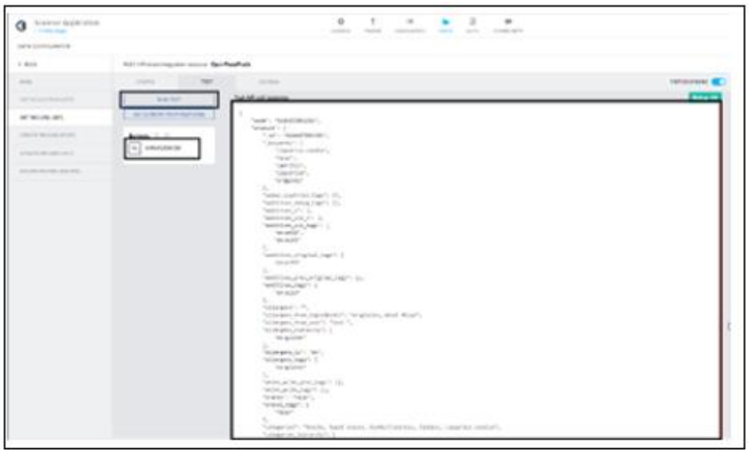
Configuring get API call: You now need to configure which information is taken from the Open Food Facts API. For your application, you need to configure this to fetch the barcode information, since that is what is being scanned with the device camera. This can be achieved using a Get Record request. To configure this, click Get Record (Get). In the Relative path field enter /product/{barcode}. Click the existing URL placeholder, (id) and then configure the following settings: Label: Barcode, Key: barcode, Value type: Text, Description: Resource ID to retrieve, Is encoded: Enabled, Is static: Disabled, Is optional: Disabled.


Data
No systems integrated
CREATE DATA ENTITY
REST API direct integration
OData integration
Google Firebase / Cloud Firestore
SAP BTP Destination REST API integration
Marketplace search

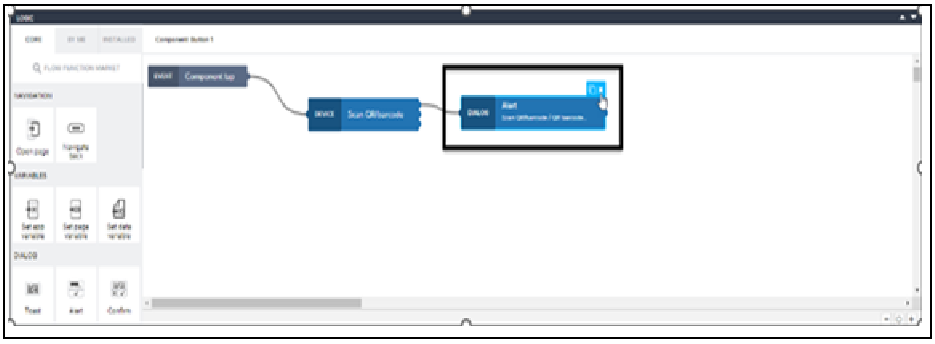
To now test that the data resource is configured properly and able to fetch information, click the Test tab. Under the Barcode placeholder, click currently bound to static text.



Add the text you want to as key then click on Run Test. The test now runs, displaying a test API call response. In this response, you can see information about the product. This includes the product categories, allergen information, and the brand that manufactured the product. Click on the set schema from the response button and click save.

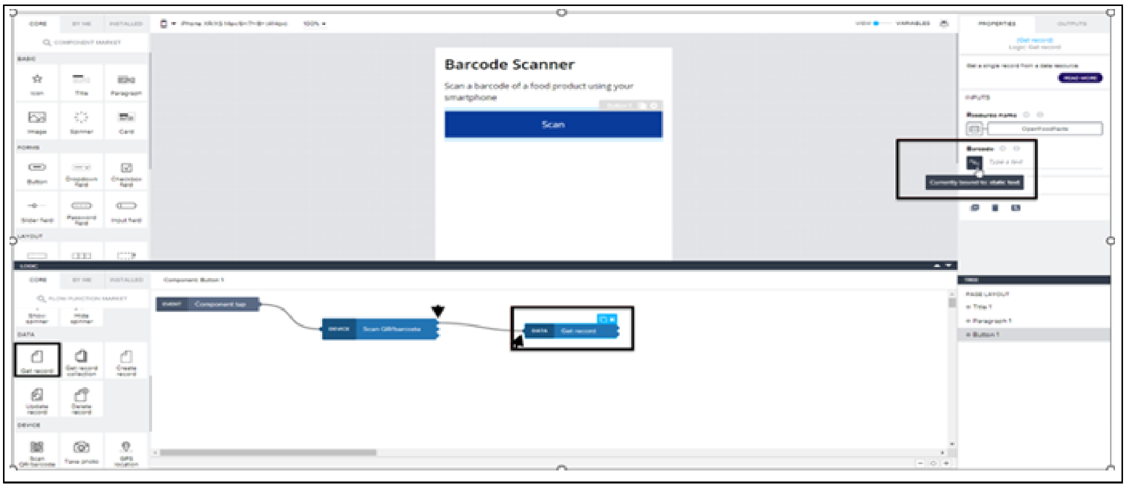


Fetch Data from API to Your Application: Open your draft application in your Composer account, displaying your barcode scanner app. As you no longer need your application to send an alert, as this was just used as a test, you need to start by removing the alert component in your logic flow. To do this, click your Scan button and then click Show Logic for Button 1. Remove the Alert component from your logic panel, as this is no longer needed.

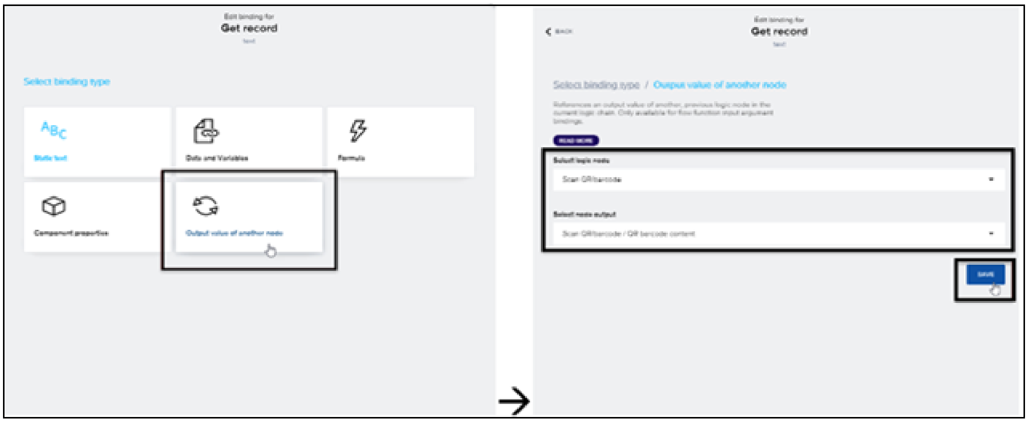


.

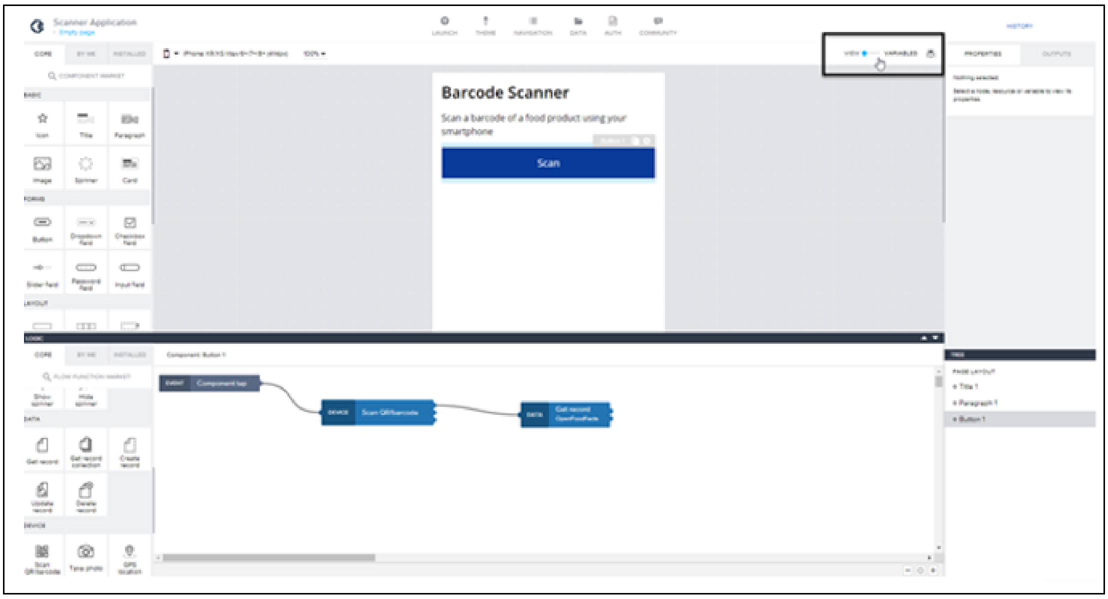
You now need to add your new logic flow for what should happen after the barcode has been scanned. To do this, using the core logic options, scroll down to Data – Get Record and then drag and drop this into the logic editor. Once the logic flow is set, you need to bind the information to the output of the scanner node. To do this, select the Get Record element and, using the properties panel, click Currently bound to Static text, opening the binding options screen.



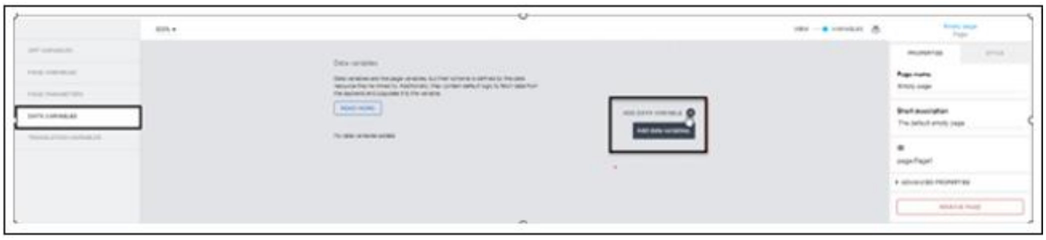
Click the Output value of another node. Configure the binding to the following: Select logic node – Scan QR/barcode, Select node output – Scan QR/barcode / QR barcode content. Then click Save.



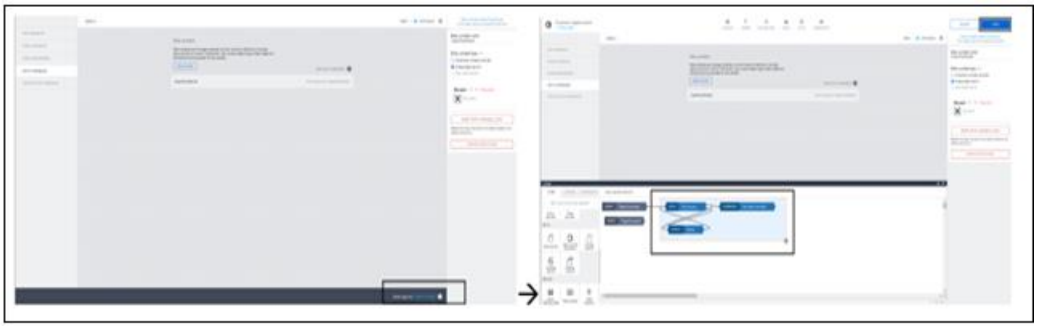
Add data variables: You now need to configure your application to store the data it receives. To do this, you need to add data variables. To do this, switch to the Variables view.



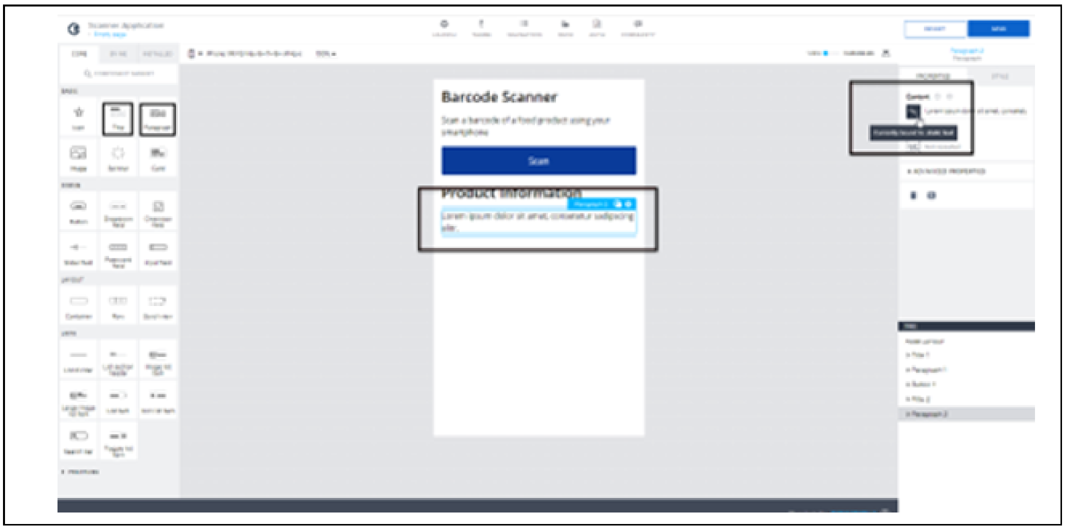
Click on the data variable, then click on the add data variables as depicted in below screenshot.

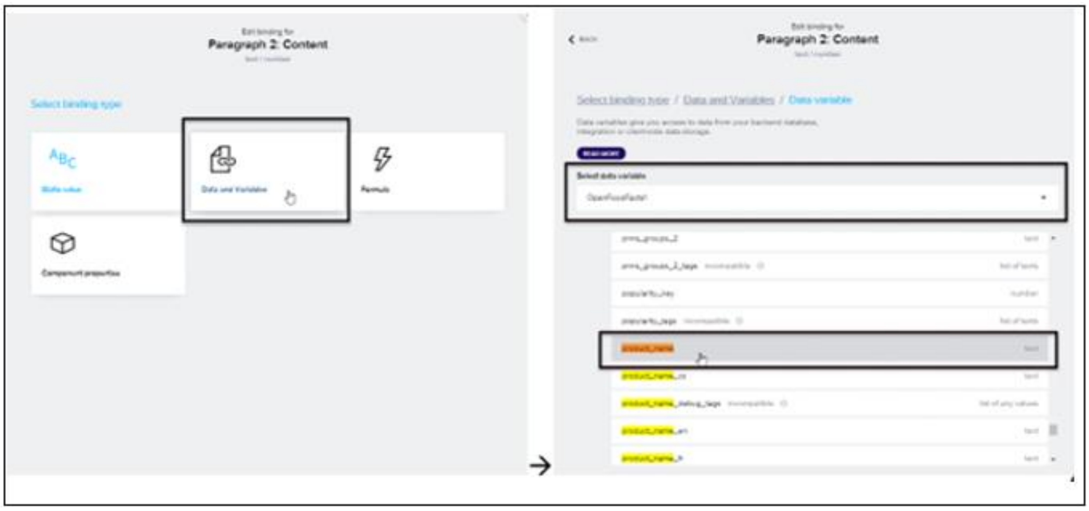


Data variables come with default logic that fetches new information every five seconds, however your app should only fetch information when a barcode is scanned. As a result, you need to remove the default logic. To do this, click Show Logic for Empty Page. Then delete the default logic by highlighting it and pressing the delete button on the keyboard. Click Save.

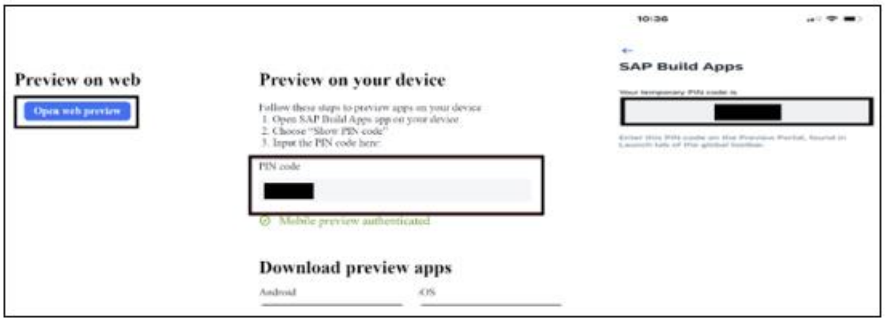


Store API data in the data variable: Click View to switch back to your application interface view. From here, you will need to add the final piece to your logic flow, storing the retrieved data in the data variable. Click your Scan button to open the relevant logic panel. Using the core logic options, scroll down to Variables – Set Data Variables and then drag and drop this into the logic editor. Now you must store the data you just retrieved in the data variable. The following provides 2 ways to do this. The first way is the standard way, but for some people, this may cause the SAP Build Apps editor to hang (you can click to exit). So we have provided a second way to store the data using a formula. Select Output value of another node and then choose the following: Select logic node: Get record Select node output: Record. Instead, you can do the same thing with a formula. Most, if not all, bindings can be done with the UI or manually with a formula. Select Formula, and then enter for the formula the following: outputs ["Get record"]. record Click Save to save this logic (no matter how you entered it).


Barcode Scanner
Scan a barcode of a food product using your smartphone
Scan
Product Information
Lorem ipsum dolor sit amet, consetetur sadipscing elitr



Launching/Previewing the application: Open the Launch tab.
Open the SAP Build Apps preview app on your mobile device and click SAP Build Apps. This will display a PIN code. Use that in the web preview page to authenticate mobile preview.


Preview on web

Open web preview

Preview on your device

Follow these steps to preview apps on your device:
1. Open SAP Build Apps app on your device
2. Choose "Show PIN code"
3. Input the PIN code here:

PIN code
[redacted]

Mobile preview authenticated

Download preview apps

Android iOS

10:36

SAP Build Apps

Your temporary PIN code is
[redacted]

Enter this PIN code on the Preview Portal, found in Launch tab of the global toolbar.

2. Conclusion

This handbook offers a guide, for developers and consultants looking to make the most of SAP Build Apps for creating and deploying enterprise applications efficiently. We started by learning how to develop applications using SAP Build Apps exploring its visual development features for designing user interfaces, managing data, and implementing logic without coding. This approach allows both developers and business users to prototype and refine applications quickly.

The next important step was configuring SAP Build Apps within SAP BTP. We discussed setting up entitlements, configuring services, and handling connections to systems like SAP S/4HANA and SuccessFactors to ensure integration and optimal performance within the SAP ecosystem. We also emphasized the use of Composer as a tool in application development. The Composers drag and drop functionality makes it easy to create applications with effort catering to users, with different levels of technical expertise. Lastly, we covered launching and previewing applications to facilitate testing and deployment.
By checking your applications you can. Fix any issues before making them available, to users ensuring a seamless user experience. The process of deploying to SAP BTP was also covered, emphasizing how to make your apps accessible to end users.

In summary, SAP Build Apps on SAP BTP provides a user platform for creating, setting up, and launching business applications. This guide has offered insights and practical tips, for unlocking the potential of SAP Build Apps empowering developers and consultants to deliver top - notch solutions that add value to businesses. By adhering to the advice and recommended strategies laid out in this guidebook you can guarantee app development and deployment within the SAP environment.

Declarations

Ethics approval and consent to participate: Not Applicable
Consent for publication: All authors have consent to submit this paper to the Journal of Cloud Computing. Also, we confirm that this paper or any part of this paper was not submitted anywhere.

Availability of data and materials: Not Applicable

Competing interests: Not Applicable

Funding: Not Applicable

References

[1] "Low - Code App Development & Automation Solutions | SAP Build, " SAP. https: //www.sap. com/products/technology - platform/low - code. html

[2] “Compose and Automate with SAP Build the No - Code Way | SAP Learning, ” learning. sap. com. https: //learning. sap. com/learning - journeys/compose - and - automate - with - sap - build - the - no - code - way

[3] “SAP Build Apps | AppGyver | Visual Low - Code App Builder, ” SAP. https: //sap. com/products/technology - platform/low - code - app - builder. html?url_id=text - glo - 404 - reclink

[4] “SAP Business Accelerator Hub, ” hub. sap. com. https: //hub. sap. com/build

[5] “Business Process Automation Software | Low Code | SAP, ” www.sap. com. https: //www.sap. com/products/technology - platform/process - automation. html

[6] “SAP Build Work Zone | Digital Workplace Experience, ” SAP. https: //www.sap. com/products/technology - platform/workzone. html

[7] “SAP Help Portal, ” help. sap. com. https: //help. sap. com/docs/build - apps/service - guide/what - is - sap - build - apps

[8] “SAP Help Portal, ” help. sap. com. https: //help. sap. com/docs/build - apps/service - guide/right - way - to - sap - build - apps

[9] “Creating Processes and Automation with SAP Build Process Automation | SAP Learning, ” learning. sap. com. https: //learning. sap. com/learning - journeys/create - processes - and - automations - with - sap - build - process - automation

[10] “Mission, ” SAP. https: //developers. sap. com/mission. appgyver - low - code. html